

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

IAGO DE CASTRO ANDRADE
Orientador: Pablo Luiz Araújo Munhoz

**OTIMIZAÇÃO DA ESCALA DE MOTORISTAS DO TRANSPORTE
PÚBLICO: UMA ABORDAGEM DE GERAÇÃO DE COLUNAS**

Ouro Preto, MG
2024

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS
DEPARTAMENTO DE COMPUTAÇÃO

IAGO DE CASTRO ANDRADE

**OTIMIZAÇÃO DA ESCALA DE MOTORISTAS DO TRANSPORTE PÚBLICO: UMA
ABORDAGEM DE GERAÇÃO DE COLUNAS**

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal de Ouro Preto como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Pablo Luiz Araújo Munhoz

Ouro Preto, MG
2024



FOLHA DE APROVAÇÃO

Iago de Castro Andrade

Otimização da escala de motoristas do transporte público: uma abordagem de Geração de Colunas

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 14 de Novembro de 2024.

Membros da banca

Pablo Luiz Araújo Munhoz (Orientador) - Doutor - Universidade Federal de Ouro Preto
Matheus Nohra Haddad (Examinador) - Doutor - Universidade Federal de Ouro Preto - Campus João Monlevade
Bárbara Letícia Rodrigues Milagres (Examinadora) - Mestranda - PPGCC/DECOM/UFOP

Pablo Luiz Araújo Munhoz, Orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 14/11/2024.



Documento assinado eletronicamente por **Pablo Luiz Araujo Munhoz, PROFESSOR DE MAGISTERIO SUPERIOR**, em 14/11/2024, às 11:24, 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 **0811450** e o código CRC **80A7F76A**.

Aos meus familiares que, sob muito sol, permitiram que eu caminhasse à sombra.

Agradecimentos

Aos meus pais, pela força e pela perseverança em me guiarem até aqui. Aos meus avós, pelo carinho e pelo apoio constante que sempre me ofereceram. Ao Ayrton, pela amizade que resiste ao tempo e à distância. Ao Breno, pela parceria de longa data e pela companhia que tornou tantos momentos especiais. À Larissa, pela companhia inigualável nas aventuras compartilhadas. Ao João, pela companhia que trouxe leveza aos dias. Ao Marcelo, cuja presença fez diferença em tantos momentos. Aos amigos, próximos ou virtuais, que contribuíram para minha jornada e deixaram uma parte de si em quem me tornei. Ao orientador, por ter assumido um desafio de grande proporção em um prazo tão curto e pelo suporte inigualável. E, por fim, às experiências e aos desafios que, em sua própria forma, lapidaram esta jornada.

“In darkness seek joy.
Surrender not to sadness, and see beyond despair.
Walk free, and bear the light for others to follow.”

Venat, Final Fantasy XIV

Resumo

A técnica de Geração de Colunas é comumente empregada na solução do Problema de Programação de Tripulações (PPT), que visa encontrar o número mínimo de motoristas necessário para cobrir todas as viagens previstas na tabela de horários de uma empresa de transporte público. O objetivo central deste trabalho é modelar o PPT como um problema de particionamento de conjuntos em conformidade com as necessidades e restrições das empresas de ônibus, e resolvê-lo partindo de um conjunto pré-compilado razoavelmente grande de jornadas potenciais (chamado de *shift-pool*) com conhecimento específico do problema. O algoritmo de geração de colunas descrito neste trabalho decompõe esse problema em um problema mestre e uma série de subproblemas de *pricing*. O problema mestre seleciona as melhores jornadas a partir de um conjunto de jornadas conhecidas, selecionadas do *shift-pool*, enquanto o subproblema de *pricing* amplia o conjunto de jornadas viáveis para aprimorar a solução obtida no problema mestre. O algoritmo proposto foi então testado com dados reais provenientes de empresas de ônibus que operam na região metropolitana de Belo Horizonte, concluindo que o modelo proposto, com o uso de geração de colunas e um *shift-pool* otimizado, reduziu de forma eficiente o número de jornadas necessárias, atendendo às restrições legais e operacionais e produzindo escalas de trabalho ajustadas e econômicas para as empresas locais.

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

Abstract

The Column Generation technique is commonly employed in solving the Crew Scheduling Problem (CSP), which aims to find the minimum number of drivers required to cover all scheduled trips in the timetable of a public transportation company. The central objective of this work is to model the CSP as a set partitioning problem in accordance with the requirements and constraints of bus companies. The model is solved starting from a reasonably large pre-compiled set of potential journeys (referred to as the *shift-pool*) with specific knowledge of the problem. The column generation algorithm described in this work decomposes this problem into a master problem and a series of pricing subproblems. The master problem selects the best journeys from a set of known feasible journeys, selected from the *shift-pool*, while the pricing subproblem expands the set of feasible journeys to enhance the solution obtained in the master problem. The proposed algorithm was then tested with real data from bus companies operating in the metropolitan area of Belo Horizonte, concluding that the proposed model, utilizing column generation and an optimized *shift-pool*, efficiently reduced the number of required shifts, meeting legal and operational constraints and producing well-adjusted and cost-effective work schedules for local companies.

Keywords: Crew Scheduling. Column Generation. Linear and Integer Optimization. Operations Research.

Lista de Ilustrações

Figura 3.1 – Representação gráfica das tarefas ordenadas em jornadas.	11
Figura 5.1 – Tempo médio para resolver x quantidade de tarefas	24

Lista de Tabelas

Tabela 2.1 – Resumo da Revisão Bibliográfica	9
Tabela 3.1 – Tabela de Tarefas	10
Tabela 5.1 – Resultados do PPT para a geração de colunas partindo do <i>shift-pool</i> com custo 600 para jornadas do tipo dupla pegada	22
Tabela 5.2 – Resultados do PPT para a geração de colunas partindo do <i>shift-pool</i> com custo 5.000 para jornadas do tipo dupla pegada	23
Tabela 5.3 – Resultados Seleccionados dos Experimentos para Viagens de Ônibus aos Domingos	25
Tabela 5.4 – Variação Percentual dos Resultados entre Custos de Dupla Pegada (600 e 5.000)	26

Lista de Abreviaturas e Siglas

ALNS	<i>Adaptive Large Neighborhood Search</i>
CSP	<i>Crew Scheduling Problem</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
GLS	<i>Guided Local Search</i>
LAHC	<i>Late Acceptance Hill Climbing</i>
PPT	Problema de Programação de Tripulações
PPV	Problema de Programação de Veículos
RCSPP	<i>Resource Constrained Shortest Path Problem</i>
RMP	<i>Restricted Master Problem</i>
SP	<i>Shift-pool</i>
VND	<i>Variable Neighborhood Descent</i>

Sumário

1	Introdução	1
1.1	Justificativa	2
1.2	Objetivos	2
1.3	Organização do Trabalho	3
2	Revisão Bibliográfica	4
2.1	Fundamentação Teórica	4
2.2	Trabalhos Relacionados	5
3	Descrição do Problema	10
3.1	Restrições	11
3.2	Modelo Matemático	12
3.2.1	Parâmetros e Conjuntos	12
3.2.2	Variáveis de Decisão	13
3.2.3	Função Objetivo	14
3.2.4	Conjunto de Restrições	14
4	Metodologia	17
4.1	Conjunto de Colunas Inicial Promissor	17
4.2	Geração de Colunas	18
4.2.1	Problema Mestre Restrito	19
4.2.2	Problema de <i>Pricing</i>	19
5	Resultados	21
5.1	Resultados para a Geração de Colunas partindo do Conjunto de Colunas Inicial Promissor	21
5.2	Análise Comparativa dos Resultados	25
6	Conclusão e Trabalhos Futuros	28
	Referências	29

1 Introdução

O aumento da população global nas grandes cidades tem impulsionado a necessidade urgente de sistemas de transporte público mais eficientes. Esta pressão destaca a importância de soluções que reduzam tempos de viagem, congestionamentos, acidentes de trânsito e emissões poluentes (Perumal, 2004). No enfrentamento desse desafio, os sistemas de transporte público se deparam com o dilema de fornecer um serviço de alta qualidade, economicamente viável para a administração pública e acessível para os usuários. A busca pela sustentabilidade não apenas requer um serviço atrativo para usuários habituais, mas também para os não cativos, incentivando a escolha do transporte público sobre o transporte individualizado (Ibarra-Rojas *et al.*, 2015). A escolha pelo transporte público não só reduziria congestionamentos e emissões, mas também manteria custos acessíveis, especialmente para a população de baixa renda que depende do transporte público para o deslocamento ao trabalho, aos locais de estudo e de lazer. Esses desafios são agravados pela necessidade de subsídios governamentais, dada a competição orçamentária com serviços como educação, saúde e segurança pública, por exemplo (Jalon; Lara; Gelashvili, 2019).

Conforme destacado por Perumal *et al.* (2019), o planejamento operacional desses sistemas envolve a resolução de diversos subproblemas encadeados que, devido a sua complexidade intrínseca, tornam inviável uma abordagem integrada em um único passo. Esses subproblemas englobam: *i*) a definição de *rotas* (linhas) a serem percorridas, *ii*) a elaboração da *tabela de horários* de cada rota, *iii*) a *programação dos veículos* que realizarão as viagens, *iv*) a *programação das tripulações*, ou seja, a definição das jornadas diárias das tripulações que conduzirão os veículos e *v*) o *rodízio das tripulações* ao longo do mês, contemplando os descansos mínimo requerido pela legislação trabalhista. A administração pública define as rotas com base na infraestrutura, serviço desejado e demanda, construindo também a tabela de horários de cada rota. Esta tabela é, então, utilizada pela empresa de transporte para planejar sua operação, definindo a frota de ônibus e a escala completa dos motoristas e cobradores, quando for o caso.

A programação de veículos refere-se à atribuição de ônibus às viagens programadas de forma que cada viagem seja coberta por um ônibus, e o objetivo é minimizar o custo operacional com base na utilização dos ônibus. Normalmente, um ônibus percorre uma sequência de viagens desde sua saída até o seu retorno à garagem. Uma jornada é definida como o trabalho realizado por um motorista ao longo de um dia, e a programação de tripulações visa encontrar um conjunto ótimo de jornadas que cubra todas as viagens dos ônibus com o menor custo possível. Normalmente existe uma tabela de horários para cada tipo de dia: dia útil, sábados e domingos. Portanto, a programação da tripulação deve ser realizada para cada tipo de dia, tendo como entrada sua respectiva tabela de horários. Dado um conjunto de jornadas para os dias úteis, os sábados e os domingos, a serem realizados ao longo de um determinado horizonte de planejamento, por

exemplo, um mês, o problema do rodízio das tripulações, ou da escala dos motoristas, tem como objetivo atribuir as jornadas diárias aos motoristas disponíveis, de tal forma que as regras de descanso entre os dias, ao longo das semanas e do mês, sejam respeitadas, visando o menor custo. Apesar de serem problemas relacionados, o planejamento das jornadas diárias e o da escala da tripulação são resolvidos de forma separada e sequencialmente, dado que um bom planejamento do primeiro implica na redução de custos, enquanto do último implica no aumento da qualidade de vida dos funcionários (Valdes, 2010).

Restrições legais e operacionais, decorrentes da legislação trabalhista, das convenções coletivas e das regras específicas da empresa, são cruciais e devem ser atendidas. Entre as regras mais comuns estão o tempo contínuo de direção, que determina por quantos minutos um motorista pode dirigir sem descanso. A quantidade de minutos de descanso geralmente depende dos minutos dirigidos, podendo incluir pausas proporcionais a este tempo. Outras regras comuns incluem o tempo de descanso entre um dia e outro, a folga nos dias de fim de semana, o retorno à garagem, limites diários e mensais de horas de direção e de horas extras, descanso diário mínimo e pausas para refeições (Valdes, 2010).

1.1 Justificativa

Este trabalho surgiu da necessidade premente de aprimorar os sistemas de transporte público frente à crescente pressão decorrente do aumento populacional nas cidades globais. A busca por soluções eficientes para mitigar desafios como tempos de viagem prolongados, congestionamentos, acidentes e emissões poluentes é crucial. O objetivo deste estudo foi compreender, dentro dos limites considerados, os métodos de otimização da alocação de recursos, considerando não apenas a eficiência operacional, mas também aspectos como acessibilidade, sustentabilidade e equidade financeira.

Neste trabalho foi realizada uma implementação com a devida complexidade requerida na resolução do clássico Problema de Programação de Tripulações. Desta forma, foi uma oportunidade para aprimorar os conhecimentos teóricos e computacionais relativos à área de otimização computacional. A condução dos experimentos computacionais, baseados em dados reais de empresas brasileiras, permitiu vivenciar as dificuldades de resolver problemas inerentes ao dia-a-dia das empresas do setor.

1.2 Objetivos

Este trabalho teve como objetivo solucionar o desafiante Problema de Programação de Tripulações do sistema de transporte público, com o intuito de atender às restrições legais e operacionais de empresas que atuam no setor. A resolução deste problema incluiu a definição das jornadas diárias de trabalho das tripulações, visando minimizar os custos operacionais, observando

rigorosamente as restrições impostas pelas leis e acordos trabalhistas, além de atender as regras operacionais estabelecidas pela empresa estudada. Devido a tais restrições, o PPT mostrou ser um problema de difícil resolução, o que requer o emprego de técnicas específicas para abordar problemas de programação inteira de grande porte.

Mais especificamente, implementou-se uma das técnicas de geração de colunas apresentadas na literatura para resolver o problema. A implementação abrangeu o método de geração de colunas que faz uso de um conjunto pré-compilado de colunas promissoras. Essa abordagem visou enfrentar o desafio de maneira abrangente e multifacetada, a fim de oferecer um comparativo com as abordagens já existentes. Também foi possível realizar um comparativo com outras abordagens da literatura, visando identificar a eficácia relativa dessas metodologias.

1.3 Organização do Trabalho

A organização do conteúdo deste trabalho é a seguinte: o Capítulo 2 aborda a revisão da literatura, fundamentando a técnica de geração de colunas e destacando trabalhos relacionados ao PPT e ao método utilizado neste trabalho. O Capítulo 3 apresenta detalhes do problema, enquanto no Capítulo 4 são discutidos modelos e heurísticas utilizados para resolver o problema. O Capítulo 5 aborda os experimentos realizados e o comparativo com outros trabalhos da literatura. A conclusão e as direções para trabalhos futuros são abordadas no Capítulo 6.

2 Revisão Bibliográfica

Nesta seção, serão abordadas as principais técnicas de otimização, como o Algoritmo Simplex e a Programação Inteira, seguidas pela técnica de geração de colunas e uma revisão de trabalhos relacionados ao Problema de Programação de Tripulação, destacando abordagens clássicas e recentes.

2.1 Fundamentação Teórica

A otimização é uma área central dentro da pesquisa operacional e visa encontrar soluções ótimas para problemas complexos que envolvem a alocação de recursos limitados. Esses problemas aparecem em diversas áreas, como planejamento de produção, escalonamento de tarefas, roteamento de veículos e design de redes. Diferentes técnicas foram desenvolvidas ao longo dos anos para lidar com a grande variedade de problemas de otimização, variando de métodos exatos a heurísticos e metaheurísticos.

Um dos marcos na história da otimização foi a criação do Algoritmo Simplex por George Dantzig em 1947 (Dantzig, 1963). O Simplex revolucionou a resolução de problemas de programação linear ao oferecer uma maneira eficiente de encontrar soluções ótimas movendo-se ao longo dos vértices do conjunto viável. Esse algoritmo tornou-se amplamente utilizado em diversas áreas da pesquisa operacional, especialmente em problemas de otimização de grande porte, como planejamento de produção e logística. No entanto, conforme os problemas de otimização tornaram-se mais complexos, novas abordagens foram desenvolvidas para complementar e, em alguns casos, superar as limitações do Simplex.

Winston (2004) aborda diversos métodos avançados que são empregados para resolver problemas de otimização, além do Simplex. Um exemplo é a programação inteira, amplamente utilizada quando as variáveis de decisão precisam assumir valores inteiros. Nesse contexto, métodos como o *branch-and-bound* são utilizados para explorar o espaço de soluções de maneira eficiente, evitando a avaliação de todas as combinações possíveis. Além disso, o autor explora o uso de heurísticas e metaheurísticas, como a busca tabu e os algoritmos genéticos, que são capazes de fornecer soluções aproximadas para problemas complexos onde a busca por uma solução exata seria computacionalmente inviável.

Entre as técnicas de otimização aplicáveis a problemas de grande escala, a geração de colunas destaca-se por sua eficácia em lidar com grandes conjuntos de variáveis. Enquanto o Simplex e o *branch-and-bound* são métodos poderosos, eles podem enfrentar dificuldades quando o problema envolve um número muito elevado de variáveis (Winston, 2004).

A geração de colunas é uma técnica amplamente utilizada para resolver problemas de

grande escala, como o PPT, que podem ser modelados como problemas lineares ou inteiros. Essa abordagem tem como objetivo otimizar problemas através da decomposição, dividindo-os em um problema mestre e um subproblema, e resolvendo-os iterativamente. Dantzig e Wolfe (1960) introduziram a técnica de geração de colunas como parte de seu método de decomposição, no qual a solução de um problema de otimização é dividida em dois: um problema mestre, que contém um subconjunto das variáveis originais (colunas), e um subproblema de precificação, que busca encontrar novas variáveis (colunas) que possam melhorar a solução atual. O problema mestre é resolvido de maneira iterativa, com a adição de colunas conforme necessário, com o objetivo de melhorar a solução.

Uma característica essencial da geração de colunas é que ela é frequentemente usada para resolver problemas de otimização com grandes quantidades de variáveis, como o Problema de Corte de Estoque e o PPT. A cada iteração, a solução é refinada ao se adicionar colunas (variáveis) ao problema mestre, as quais são geradas a partir da resolução do subproblema. Esse processo continua até que nenhuma coluna adicional melhore a solução (Desrosiers; Lubbecke, 2006).

A decomposição de Dantzig-Wolfe, que é um dos pilares da geração de colunas, facilita a resolução de problemas com estruturas complexas e restrições, permitindo que eles sejam tratados de forma eficiente e otimizados em grande escala (Dantzig; Wolfe, 1960). Outro aspecto importante é que, além da decomposição de Dantzig-Wolfe, a geração de colunas pode ser combinada com a relaxação lagrangeana e técnicas de *branch-and-bound*, formando um framework robusto para lidar com problemas que exigem soluções inteiras (Desrosiers; Lubbecke, 2006).

Na prática, a geração de colunas é aplicada frequentemente em problemas que envolvem a minimização de custos e a alocação eficiente de recursos, como em problemas de roteamento de veículos e programação de tripulação. O processo de geração de colunas envolve um problema mestre restrito (*Restricted Master Problem* - RMP), onde são otimizadas apenas algumas variáveis de decisão inicialmente selecionadas. Novas colunas são inseridas ao resolver o subproblema de precificação, que identifica variáveis com custos reduzidos negativos, indicando que podem melhorar a solução (Desrosiers; Lubbecke, 2006).

2.2 Trabalhos Relacionados

O Problema de Programação de Tripulação (PPT) do sistema de transporte público é um dos desafios mais significativos na logística de transporte, envolvendo a criação das jornadas diárias de trabalho das tripulações que devem conduzir uma dada frota (Huisman, 2004). O estudo deste problema é de grande importância para o setor de transporte público, uma vez que os custos relacionados à remuneração das tripulações têm um grande impacto na contabilidade da empresa, estimado em 60% dos seus custos totais (Perumal *et al.*, 2019). Além disso, é fundamental destacar que o PPT é classificado como NP-difícil, mesmo quando são consideradas apenas as restrições relacionadas ao tempo total de trabalho e aos intervalos de descanso e alimentação

das tripulações (Fischetti; Martello; Toth, 1987; Fischetti; Martello; Toth, 1989). Um problema NP-difícil é aquele para o qual não existe algoritmo polinomial que o resolva, ou seja, o tempo de resolução cresce exponencialmente com o tamanho dos dados de entrada, impossibilitando a sua execução mesmo para problemas de médio porte.

O PPT é a etapa subsequente ao Problema de Programação dos Veículos (PPV), na programação das atividades diárias de uma empresa de transporte. Enquanto o PPV se concentra em detalhar o tamanho da frota utilizada e atribuir as viagens a serem realizadas por cada veículo em operação, o PPT tem como objetivo alocar as tarefas de condução dos ônibus aos motoristas e cobradores, conhecidos como tripulações. Uma solução do PPT deve fornecer as jornadas de trabalho das tripulações, garantindo que as viagens dos veículos em operação sejam realizadas de forma eficiente, visando à minimização dos custos operacionais (Cunha, 2021). O PPT é um problema amplamente abordado na literatura, uma vez que uma programação eficaz das jornadas das tripulações resulta em economias significativas para as empresas do setor. No entanto, é importante ressaltar que o escalonamento eficiente das tripulações representa um desafio significativo devido à complexidade envolvida na sua realização.

Uma das abordagens propostas para resolver problemas como este, que envolvem um grande número de variáveis, é a aplicação do método Simplex associado à técnica de geração de colunas. O conceito central dessa abordagem, proposta inicialmente por Dantzig e Wolfe (1960), consiste em resolver uma sequência de problemas simplificados, sendo que cada problema simplificado contém um conjunto restrito de variáveis, associadas às colunas. Ao resolver um problema simplificado, um novo conjunto de colunas é gerado utilizando as informações duais da solução ótima corrente. O algoritmo de geração de colunas é interrompido quando não for possível gerar novas colunas capazes de melhorar a solução corrente. Nesse ponto, a solução ótima do problema simplificado corresponde à solução ótima do problema original. O problema simplificado é referido como o “problema mestre”, enquanto o processo de gerar um novo conjunto de colunas é conhecido como o “problema de precificação” ou *pricing* (Huisman, 2004).

Desrochers e Soumis (1989) foram pioneiros ao utilizar o algoritmo de geração de colunas aliado ao algoritmo de programação inteira *branch and bound*, adaptado para refletir as condições do nó atual na árvore de busca. A geração de colunas foi dividida em dois componentes principais, sendo que o primeiro abordou um problema de cobertura de conjunto, escolhendo uma programação a partir de dias de trabalho viáveis conhecidos, e o segundo tratou de otimizar um desafio crucial na área de logística, conhecido como *Resource Constrained Shortest Path Problem* (RCSPP) ou, problema do caminho mínimo com restrições de recursos. O objetivo foi propor novas jornadas de trabalho viáveis para aprimorar a solução atual. Os resultados apresentados foram superiores aos obtidos pela heurística de *runcutting*, o método HASTUS e o Problema de Cobertura de Conjuntos até então (Desrochers; Soumis, 1989).

Uma alternativa para facilitar a resolução da geração de todos os turnos potenciais na geração de colunas foi a representação desse subproblema como um RCSPP, conforme modelado

por Chen e Shen (2013) como um problema de Programação Linear Inteira. O problema do caminho mínimo com restrições de recursos é NP-difícil e contribui para a lenta convergência da geração de colunas. Nesse sentido, um conjunto considerável de jornadas potenciais de qualidade foi compilado antecipadamente, chamado de *shift-pool*. Durante a geração de colunas, o RCSPP só é acionado novamente quando não há mais colunas com custo reduzido negativo no conjunto gerado inicialmente. Os resultados computacionais obtidos pelo algoritmo são promissores, destacando-se pela significativa economia de tempo computacional alcançada.

Considerando as limitações associadas ao uso da técnica de geração de colunas para resolver problemas de grande escala, Silva e Cunha (2010) apresentaram um modelo que se fundamenta na metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*). Esta abordagem inclui uma busca local conduzida pelo método de Busca em Vizinhança de Grande Porte, possibilitando a reorganização das atribuições de tarefas e a realocação de recursos, independentemente do número de tripulações envolvidas. Posteriormente, os autores resolveram o problema empregando a metaheurística *Iterated Local Search*, combinada com a técnica de busca supracitada. Isso resultou na capacidade de realocação e troca de tarefas de forma mais direcionada e menos aleatória entre as equipes, melhorando a eficácia do processo (Silva; Reis, 2014).

No trabalho de Souza (2014), foram desenvolvidos métodos fundamentados em um modelo de programação linear inteira que se caracteriza por gerar cada jornada de forma direta, com base nas tarefas que requerem alocação. Adicionalmente, a investigação empregou a metaheurística *Late Acceptance Hill Climbing* (LAHC) para lidar com desafios de maior complexidade. Notavelmente, um método híbrido, ou seja, uma *Matheurística*, foi concebida, combinando uma abordagem exata com a metaheurística LAHC, o que permitiu aprimorar as soluções provenientes da metaheurística e, conseqüentemente, resultar na redução dos custos associados às jornadas programadas.

O estudo conduzido por Silva e Silva (2015) empregou a técnica *Guided Local Search* (GLS) como uma metaheurística, com a finalidade de eliminar características indesejáveis nas soluções ótimas locais, abrindo caminho para explorar amplamente o espaço de soluções e outros ótimos locais. Adicionalmente, incorporou a heurística *Variable Neighborhood Descent* como estratégia de busca local, explorando diversas estruturas de vizinhança na busca por mínimos locais. Destaca-se que essa abordagem revela-se como uma contribuição inédita para a resolução do PPT.

Constantino *et al.* (2017) propuseram um algoritmo heurístico determinístico de duas fases que utiliza problemas de atribuição múltipla originados de um modelo baseado em um grafo multipartido ponderado para instâncias grandes, sendo a maior delas com 2.300 tarefas, ou seja, conjuntos de viagens sequenciais realizadas por um mesmo veículo. Na primeira fase, o algoritmo cria uma solução inicial viável resolvendo diversos problemas de atribuição. Na segunda fase, busca-se aprimorar a solução por meio de dois procedimentos distintos: um deles divide o

conjunto de tarefas em tarefas parciais que são recombinadas, enquanto o outro visa otimizar tarefas longas, eliminando o tempo extra e incorporando-o a outras jornadas. Os resultados experimentais demonstraram competitividade na resolução de instâncias extensas.

No trabalho conduzido por Martins (2017), foi utilizada a heurística de busca local denominada *Adaptive Large Neighborhood Search* (ALNS), cujo foco reside na minimização dos custos fixos e variáveis inerentes a uma programação exaustiva das equipes, assegurando o cumprimento integral das restrições estipuladas. Os custos fixos estão intrinsecamente associados à determinação do número de jornadas requeridas, ao passo que os custos variáveis são calculados com base nas horas extras, nas horas ociosas e na formação de jornadas do tipo dupla pegada. A ALNS inicia sua busca a partir de uma solução que seja viável e, conseqüentemente, emprega diversos métodos de “destruição” e “reparo” na solução vigente, com a finalidade de aprimorar a qualidade da melhor solução conhecida.

O experimento de Perumal *et al.* (2019) combinou técnicas de programação matemática com metaheurísticas para alcançar soluções de alta qualidade em um tempo computacional razoável. Essa abordagem utilizou um método de busca em vizinhança adaptativa, onde soluções vizinhas são geradas por meio de operações de destruição e reconstrução. Diversos métodos de destruição e reconstrução foram aplicados, com pesos ajustados dinamicamente com base no desempenho, permitindo explorar diferentes vizinhanças de solução durante a busca. Em comparação com os resultados obtidos pelo *solver* de referência, observaram-se melhorias de até 15%.

Cunha (2021) implementou uma estratégia de geração de colunas que incorporou heurísticas para resolver o problema de precificação no relaxamento linear do problema. Além disso, ele utilizou uma heurística de cobertura para obter limites inferiores. Os experimentos computacionais evidenciaram a alta qualidade do limite inferior obtido no relaxamento linear, bem como o potencial da heurística de cobertura para gerar soluções de qualidade, estando bem próximas à otimalidade na maioria das instâncias.

A revisão da literatura evidenciou que o PPT continua sendo um problema relevante, merecendo atenção da comunidade científica. Ele tem sido abordado de diferentes formas, seja aplicando técnicas exatas de Programação Inteira, seja por meio de metaheurísticas de otimização. Trabalhos mais recentes utilizam a matheurística para abordar o problema. É neste contexto que se insere o presente estudo, qual seja, na resolução do PPT, tendo em vista aplicar a casos reais brasileiros. Um resumo das principais abordagens discutidas nesta seção pode ser encontrado na Tabela 2.1.

Tabela 2.1 – Resumo da Revisão Bibliográfica

Autor(es)	Ano	Metodologia	Principais Resultados
Fischetti et al.	1987	Programação Inteira	Classificação do PPT como NP-difícil, mesmo com restrições de tempo de trabalho e descanso.
Desrochers et al.	1989	Geração de colunas com branch and bound	Resultados superiores a heurísticas como HASTUS, com enfoque em jornadas otimizadas com RCSPP.
Huisman et al.	2004	Geração de colunas	Aplicação da geração de colunas para o PPT, foco em otimização do custo operacional.
Silva et al.	2010	Metaheurística GRASP	Reorganização eficiente de tarefas e alocação de recursos com busca em vizinhança de grande porte.
Chen et al.	2013	Programação Linear Inteira (RCSPP)	Redução do tempo de solução com a compilação de jornadas viáveis previamente ao processo de geração de colunas.
Souza et al.	2014	Programação Linear Inteira com LAHC	Matheurística híbrida com LAHC, resultando em redução de custos operacionais.
Silva et al.	2015	Guided Local Search (GLS) e Variable Neighborhood Descent (VND)	Exploração extensiva de ótimos locais para soluções eficientes no PPT.
Constantino et al.	2017	Algoritmo heurístico determinístico de duas fases	Criação de soluções para grandes instâncias (até 2.300 tarefas), com otimização de tarefas longas.
Martins et al.	2017	Adaptive Large Neighborhood Search (ALNS)	Minimização de custos fixos e variáveis com soluções otimizadas para jornadas do tipo dupla pegada.
Perumal et al.	2019	Matheurística	Redução de até 15% no tempo computacional com técnicas de destruição e reconstrução de vizinhanças.
Cunha et al.	2021	Geração de colunas com heurísticas de precificação	Obtenção de limites inferiores de alta qualidade e soluções próximas à otimalidade.
Andrade et al.	2024	Geração de colunas partindo de um <i>shift-pool</i> pré-compilado	Geração de soluções próximas à otimalidade, com eficiência e desempenho competitivo no tempo de execução.

Fonte: Elaborado pelo autor.

3 Descrição do Problema

Para tratar o Problema de Programação da Tripulação, parte-se do escalonamento dos veículos, que pode ser obtido resolvendo o Problema de Programação de Veículos, quando as viagens são atribuídas a um conjunto de veículos, de forma a obter o menor custo em relação ao total de veículos e ao custo operacional da frota. Esse escalonamento viabiliza a criação das *tarefas*, sendo cada tarefa identificada por um conjunto de viagens sequenciais realizadas por um mesmo veículo. É importante ressaltar que as tarefas são constituídas de sequências de viagens que não permitem a troca de tripulações. As tarefas são os dados de entrada utilizados para a formação das jornadas, ou seja, as jornadas são combinações de tarefas que satisfazem as restrições do problema.

O propósito do PPT consiste em atribuir eficientemente um conjunto de tarefas às tripulações tendo em vista a minimização simultânea da quantidade de tripulantes envolvidos e do custo operacional, representados por horas extras e intervalos de ociosidade. Ao realizar esse escalonamento, é imperativo seguir rigorosamente todas as restrições impostas pela empresa, considerando, adicionalmente, as limitações estabelecidas por regulamentações legais e acordos trabalhistas.

Tabela 3.1 – Tabela de Tarefas

Tarefa	Hora de Partida	Local de Partida	Hora de Chegada	Local de Chegada	Veículo	Duração
1	00:30	GARAGEM	03:30	PAMPULHA	1	3:00
2	03:05	GARAGEM	06:00	SAVASSI	2	2:55
3	03:45	GARAGEM	06:00	BURITIS	3	2:15
4	04:00	PAMPULHA	06:00	PAMPULHA	1	2:00
5	06:10	BURITIS	08:40	GARAGEM	3	2:30
6	06:10	SAVASSI	09:00	SAVASSI	2	2:50
7	06:17	PAMPULHA	09:07	PAMPULHA	1	2:50

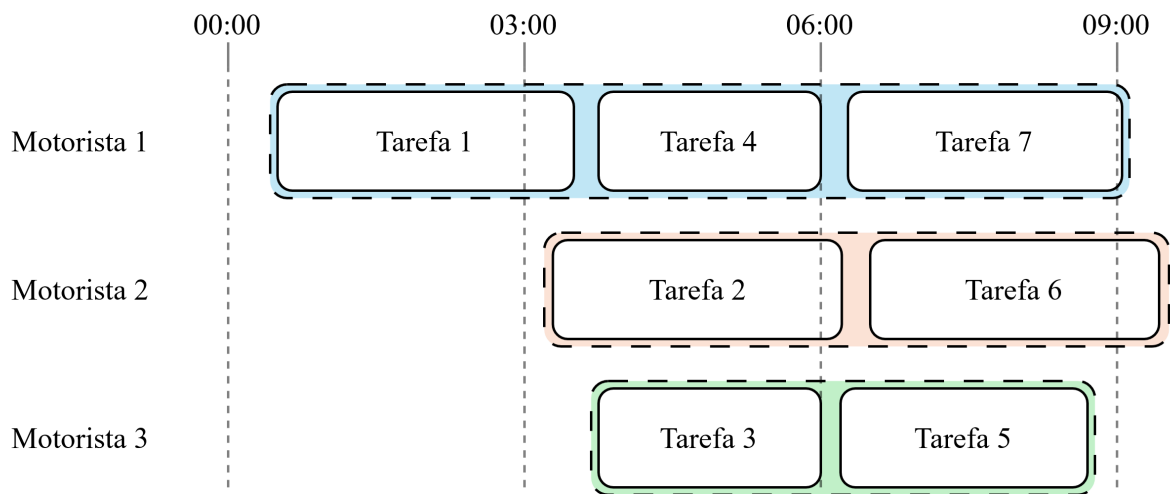
Fonte: Elaborado pelo autor.

A Tabela 3.1 fornece detalhes sobre as tarefas, incluindo informações como a hora de seu início (em horas), o ponto inicial correspondente ao local de início da tarefa, a hora de seu término (em horas), o ponto final relacionado ao local onde a tarefa é concluída, o número do veículo responsável pela execução e a sua duração em horas. Este exemplo representativo reflete as características das tarefas obtidas pelo PPV, com tratamento adicional realizado para fins de ordenação das entradas por horário de início. Para ilustrar, considere a primeira entrada na Tabela 3.1. Nessa instância, a tarefa se inicia às 00:30 horas no ponto “GARAGEM” e conclui às 03:30 horas no ponto “PAMPULHA”.

Para complementar a compreensão da Tabela 3.1, a Figura 3.1 apresenta uma representação gráfica das tarefas, onde é possível visualizar de forma mais intuitiva a formação das jornadas

de trabalho.

Figura 3.1 – Representação gráfica das tarefas ordenadas em jornadas.



Fonte: Elaborado pelo autor.

3.1 Restrições

Para que uma solução do PPT seja viável, é preciso que ela satisfaça várias restrições. Estas incluem as regulamentações legais brasileiras e as limitações específicas da empresa transportadora localizada na região metropolitana de Belo Horizonte. No escopo deste trabalho, foram consideradas as seguintes restrições:

1. Uma jornada é considerada como “dupla pegada” quando o intervalo entre duas tarefas consecutivas ultrapassa duas horas, resultando na não remuneração do motorista pelo tempo ocioso entre essas tarefas.
2. Se o intervalo entre duas tarefas quaisquer for inferior a duas horas, a jornada é classificada como “jornada corrida”, e o motorista é remunerado pelo tempo ocioso entre as tarefas.
3. O período de descanso em casa, entre o término e o início da jornada no dia seguinte, deve ser de pelo menos 11 horas contínuas para todos os motoristas. Portanto, se um motorista encerrar sua jornada às 00:00 horas, só poderá retomar o trabalho no próximo dia a partir das 11:00 horas da manhã.
4. A duração da jornada de trabalho é de 6 horas e 40 minutos, podendo ser estendida por até duas horas extras.
5. Em jornadas com duração de até 4 horas, o motorista não requer um tempo mínimo de descanso.

6. Jornadas com duração entre 4 e 6 horas exigem um descanso mínimo de 15 minutos para o motorista.
7. Jornadas com duração superior a 6 horas requerem pelo menos 30 minutos de descanso para o motorista.
8. Uma dupla pegada é dividida entre a 1ª e a 2ª pegada, sendo que cada pegada não pode exceder 6 horas de duração, sem tempo de descanso previsto nesse intervalo.
9. Nas jornadas corridas, não são permitidas trocas de pontos. Nas duplas pegadas, a troca de pontos pode ocorrer, mas somente entre a 1ª e a 2ª pegada.
10. Considerando duas tarefas sequenciais i e j desempenhadas pela mesma tripulação, o horário de início da tarefa j deve ser posterior ao horário de término da tarefa i , de forma que as tarefas não se sobreponham. Além disso, o local de partida da tarefa j deve coincidir com o local de chegada da tarefa i .
11. Cada jornada pode ter, no máximo, uma troca de veículo.

3.2 Modelo Matemático

Nesta seção apresenta-se o modelo matemático utilizado para a resolução do Problema de Programação de Tripulação, originalmente proposto por Souza (2014). A seguir, são detalhadas as variáveis de decisão, as restrições e a função objetivo que compõem o modelo.

3.2.1 Parâmetros e Conjuntos

Considere qt como o parâmetro de entrada que define a quantidade de tarefas no problema. Assim, tem-se o conjunto de tarefas T , onde $|T| = qt$. O parâmetro qj corresponde ao número máximo de jornadas na solução, gerando o conjunto de possíveis jornadas J , com $|J| \leq qj$. O número máximo de tarefas que podem ser atribuídas a uma única jornada é dado por $MaxTar$. Além disso, uma constante suficientemente grande M (Big M) é empregada na modelagem de algumas restrições.

Os parâmetros $TempMaxJ$ e $TempMaxHE$ representam, respectivamente, a duração máxima permitida para uma jornada de trabalho e o tempo máximo de horas extras que podem ser alocadas em uma jornada. Os custos associados ao modelo são dados pelos parâmetros $CustoJ$, $CustoHE$, $CustoOC$ e $CustoDP$, que representam, respectivamente, o custo fixo de uma jornada, o custo devido às horas extras, o custo relacionado ao tempo ocioso e o custo adicional para jornadas do tipo dupla pegada.

Outros parâmetros importantes incluem $QuantMaxDP$, que indica a quantidade máxima de duplas pegadas permitidas na solução, e $QuantMaxTV$, que limita o número de trocas de

veículos por jornada. O parâmetro $InterMinDP$ define o intervalo mínimo necessário entre duas tarefas para que caracterizem uma jornada do tipo dupla pegada, enquanto $InterMinT$ corresponde ao tempo mínimo de descanso ou alimentação necessário para jornadas do tipo pegada simples. Ressalta-se que todos os parâmetros mencionados possuem valores inteiros e positivos.

Para cada tarefa $t \in T$, são definidos os seguintes atributos: st_t representa o horário de início da tarefa, et_t o horário de término, sl_t o terminal de saída, el_t o terminal de chegada, e vc_t o veículo que realiza a tarefa.

Dadas duas tarefas $t_1, t_2 \in T$, o parâmetro $PosTV_{t_1t_2}$ identifica a possibilidade de troca de veículos entre elas. Este valor é definido como $PosTV_{t_1t_2} = 1$ se $vc_{t_1} \neq vc_{t_2}$, $el_{t_1} = sl_{t_2}$ e $st_{t_2} - et_{t_1} > 0$; caso contrário, $PosTV_{t_1t_2} = 0$.

Analogamente, $PosDP_{t_1t_2}$ identifica a possibilidade de uma jornada do tipo dupla pegada entre as tarefas t_1 e t_2 . Este parâmetro assume o valor $PosDP_{t_1t_2} = 1$ se $st_{t_2} - et_{t_1} > InterMinDP$; caso contrário, $PosDP_{t_1t_2} = 0$.

3.2.2 Variáveis de Decisão

Para modelar o PPT, além das variáveis de alocação das tarefas às jornadas, são necessárias algumas variáveis auxiliares para que sejam geradas soluções viáveis para o problema. As soluções viáveis devem satisfazer ao conjunto de restrições implementadas, que serão apresentadas nas próximas subseções. Tais restrições matemáticas são utilizadas para modelar as regras impostas para que uma jornada de trabalho seja viável.

Seja $j \in J$, onde $|J|$ é aproximadamente 30% do total de tarefas, $t \in T$ e $p \in P$, $|P| = 8$, onde J , T e P são os conjuntos de jornadas, tarefas e posições (correspondendo à ordem em que cada tarefa é alocada). Para cada jornada, uma tarefa será alocada a uma posição, sempre em ordem crescente. Assim, a variável binária x_{jtp} assume valor 1 se a tarefa t for alocada à jornada j na posição p . Vale ressaltar que esta é a variável que apresenta a solução do problema.

A variável y_j é utilizada para identificar se a jornada j tem alguma tarefa alocada a ela, recebendo o valor 1; caso contrário, seu valor será 0. A variável ini_j receberá o valor do horário de início da jornada j , fim_j receberá o valor do horário de término da jornada j e OC_{int_j} terá o valor do tempo total de ociosidade interna, ou seja, a soma dos intervalos ociosos entre as tarefas atribuídas à jornada j . A variável auxiliar OC_{ext_j} receberá o tempo de ociosidade externa, quando houver, correspondente ao intervalo que falta para completar a carga horária normal de uma jornada de trabalho. Por fim, Qnt_{HE_j} receberá a quantidade de horas extras da jornada, se existirem.

Considerando $j \in J$, $t_1, t_2 \in T$ e $p_1, p_2 \in P$, a variável $TrocaV_{jt_1t_2p_1p_2}$ tem como objetivo verificar se existe uma troca de veículo entre as tarefas t_1 e t_2 , localizadas nas posições p_1 e p_2 da jornada j . As variáveis $DP_{jt_1t_2p_1p_2}$ e $Dupla_j$ são utilizadas, respectivamente, para

verificar as posições e marcar se uma jornada é do tipo dupla pegada. A variável $Desc_DP_j$ contém a amplitude do intervalo não remunerado para jornadas do tipo dupla pegada. Esse valor é descontado da remuneração da jornada.

3.2.3 Função Objetivo

Duas funções objetivo são propostas neste trabalho. A primeira, dada pela expressão (3.1), minimiza o número de duplas pegadas, sem utilizar a restrição (3.15) para limitar o número máximo de jornadas do tipo dupla pegada. A segunda função objetivo, expressa em (3.2), não minimiza o número de duplas pegadas, mas limita a quantidade de jornadas deste tipo considerando a restrição (3.15). Ambas as funções são apresentadas abaixo e foram testadas separadamente.

$$f_{o1} = \sum_{j \in J} \left(CustoSJ \cdot y_j + CustoSOC \cdot (OC_ext_j + OC_int_j) + CustoSHE \cdot QntSHE_j + CustoSJP \cdot Dupla_j \right) \quad (3.1)$$

$$f_{o2} = \sum_{j \in J} \left(CustoSJ \cdot y_j + CustoSOC \cdot (OC_ext_j + OC_int_j) + CustoSHE \cdot QntSHE_j \right) \quad (3.2)$$

Nas expressões (3.1) e (3.2), $CustoSJ$ refere-se ao custo fixo de cada jornada, $CustoSOC$ é o custo de cada hora ociosa, $CustoSHE$ é o custo de cada hora extra e $CustoSJP$ é o custo associado a cada jornada do tipo dupla pegada, quando essa componente faz parte da função objetivo. Dessa forma, minimizam-se os custos fixos, as horas extras, as horas ociosas e as duplas pegadas (quando consideradas) em uma solução.

3.2.4 Conjunto de Restrições

As restrições do modelo foram definidas e organizadas de acordo com as características do PPT descritas anteriormente. Devido à sua quantidade e diversidade, as restrições são apresentadas em grupos, agrupadas por similaridade.

$$\sum_{t \in T} x_{jtp} \leq 1, \quad \forall j \in J, p \in P \quad (3.3)$$

$$\sum_{j \in J, p \in P} x_{jtp} = 1, \quad \forall t \in T \quad (3.4)$$

$$\sum_{t_1 \in T} x_{jt_1(p+1)} \leq \sum_{t_2 \in T} x_{jt_2p}, \quad \forall j \in J, p \in P - 1 \quad (3.5)$$

$$y_j \geq x_{jtp}, \quad \forall j \in J, t \in T, p \in P \quad (3.6)$$

As restrições acima desempenham os seguintes papéis: - A restrição (3.3) garante que, em cada posição de uma jornada, apenas uma tarefa pode ser alocada. - A restrição (3.4) assegura que cada tarefa seja alocada a exatamente uma jornada. - A restrição (3.5) estabelece que as tarefas em uma jornada devem ser atribuídas em posições consecutivas, exceto na primeira posição. - A restrição (3.6) verifica se uma jornada está sendo utilizada, atribuindo o valor 1 à variável y_j caso alguma tarefa esteja alocada à jornada j , e 0 caso contrário.

$$ini_j = \sum_{t \in T} st_t \cdot x_{jt_1}, \quad \forall j \in J \quad (3.7)$$

$$fim_j \geq et_t - M \cdot (1 - x_{jt_p}), \quad \forall j \in J, t \in T, p \in P \quad (3.8)$$

As restrições (3.7) e (3.8) definem os horários de início (ini_j) e término (fim_j) de cada jornada j , usados para garantir a consistência temporal das soluções.

$$PosDP_{t_1 t_2} + x_{jt_1 p} + x_{jt_2 (p+1)} - DP_{jt_1 t_2 p (p+1)} \leq 2, \quad \forall j \in J, t_1, t_2 \in T, p \in P - 1 \quad (3.9)$$

$$DP_{jt_1 t_2 p (p+1)} \leq x_{jt_1 p}, \quad \forall j \in J, t_1, t_2 \in T, p \in P - 1, t_2 > t_1 \quad (3.10)$$

$$DP_{jt_1 t_2 p (p+1)} \leq x_{jt_2 (p+1)}, \quad \forall j \in J, t_1, t_2 \in T, p \in P - 1, t_2 > t_1 \quad (3.11)$$

$$DP_{jt_1 t_2 p (p+1)} = 0, \quad \forall j \in J, t_1, t_2 \in T, p \in P - 1, PosDP_{t_1 t_2} = 0 \quad (3.12)$$

$$\sum_{t_1, t_2 \in T, p \in P - 1} DP_{jt_1 t_2 p (p+1)} \leq 1, \quad \forall j \in J \quad (3.13)$$

$$Dupla_j = \sum_{t_1, t_2 \in T, p \in P - 1} DP_{jt_1 t_2 p (p+1)}, \quad \forall j \in J \quad (3.14)$$

Essas restrições identificam e modelam os intervalos de jornadas do tipo dupla pegada. A restrição (3.9) verifica se duas tarefas consecutivas formam uma dupla pegada. As restrições (3.10) e (3.11) asseguram que a variável DP receba valor 1 somente quando as condições para dupla pegada forem atendidas. A restrição (3.12) impede que DP seja ativada para tarefas que não podem formar uma dupla pegada. Já (3.13) limita a uma única dupla pegada por jornada, e (3.14) verifica se uma jornada possui dupla pegada, atribuindo valor à variável $Dupla_j$.

$$\sum_{j \in J} Dupla_j \leq QntMaxDP \quad (3.15)$$

$$DescDP_j = \sum_{t_1, t_2 \in T, p \in P - 1} DP_{jt_1 t_2 p (p+1)} \cdot (st_{t_2} - et_{t_1}), \quad \forall j \in J \quad (3.16)$$

$$fim_j - ini_j - DescDP_j \leq TempMaxJ + TempMaxHE, \quad \forall j \in J \quad (3.17)$$

$$fim_j - ini_j \leq 780, \quad \forall j \in J \quad (3.18)$$

As restrições (3.15) a (3.18) limitam o número máximo de duplas pegadas, calculam o tempo de intervalo não remunerado ($DescDP_j$) e garantem que nenhuma jornada exceda a

duração máxima permitida, inclusive considerando horas extras. Por fim, (3.18) assegura que o descanso entre jornadas seja de, no mínimo, 11 horas.

4 Metodologia

O desenvolvimento deste trabalho conta com a técnica de criação de um conjunto de colunas inicial promissor apresentada por [Chen e Shen \(2013\)](#) e também com o algoritmo de geração de colunas presente no trabalho de [Lin e Hsu \(2016\)](#).

4.1 Conjunto de Colunas Inicial Promissor

Uma maneira de resolver o Problema de Programação de Tripulações por meio da geração de colunas propõe a pré-compilação de um conjunto de jornadas, denominado *shift-pool*. Este conjunto é cuidadosamente projetado para incorporar a maioria das jornadas eficientes encontradas na solução ótima, formando assim um conjunto de colunas inicial promissor. A rapidez na geração do *shift-pool* é crucial para evitar sobrecarga ao algoritmo global de geração de colunas, preservando a eficácia do processo ([Chen; Shen, 2013](#)). Baseado em conhecimento específico do problema, sabe-se que as jornadas com as seguintes características são mais eficientes:

- Conter tarefas mais longas para aumentar o tempo de trabalho ativo;
- Conter intervalos ou tempos de trocas de pontos mais curtos para reduzir o tempo de ociosidade;
- Ter jornadas corridas e jornadas do tipo dupla pegada.

De acordo com as características descritas acima e as restrições legais e operacionais impostas pela legislação trabalhista, definem-se os seguintes parâmetros para a geração do *shift-pool*:

1. *tarefa_Min*: tempo mínimo de duração de uma tarefa, $tarefa_Min \in [60, 240]$;
2. *intervalo_MaxCorrida*: tempo máximo do intervalo em uma jornada corrida, $intervalo_MaxCorrida \in [0, 120]$;
3. *trabalho_Min*: tempo mínimo de trabalho, $trabalho_Min \in [120, 520]$.

Observe que todos os intervalos dos parâmetros acima são determinados antecipadamente pelas regras trabalhistas. Antes de gerar o *shift-pool*, gera-se todas as tarefas usando um algoritmo enumerativo, que então são ordenados pelo seu horário de início em ordem não decrescente. Seja *tarefas_W* o conjunto que contém todas as tarefas e nW a cardinalidade de *tarefas_W*. Para a i -ésima *tarefa_i*, seja sT_i e eT_i os horários de início e fim, respectivamente. Denote por tw_{ij} o

tempo de trabalho da combinação da $tarafa_i$ com a $tarafa_j$, e t_{ij} o intervalo entre essas duas tarefas. O pseudocódigo para gerar o *shift-pool*, abreviado como *SP*, é dado no Algoritmo 4.1.

Algoritmo 4.1: Gerando o shift-pool

```

1  Insira o conjunto de tarefas  $tarafas_W$ ;
2  for  $i = 1$  to  $nW - 1$  do
3      if  $eT_i - sT_i \geq tarafa\_Min$  then
4          for  $j = i + 1$  to  $nW$  do
5              if  $eT_j - sT_j \geq tarafa\_Min$  then
6                   $t_{ij} \leftarrow sT_j - eT_i$ ;
7                  if  $0 < t_{ij} \leq intervalo\_MaxCorrida$  then
8                      if  $tw_{ij} \geq trabalho\_Min$  then
9                          if  $jornadaValida(i, j)$  then
10                             Combine a  $tarafa_i$  com a  $tarafa_j$  para formar uma jornada
                                corrida e adicione-a ao SP;

```

A função $jornadaValida(i, j)$ verifica a viabilidade de combinar a tarefa i com a tarefa j em conformidade com as regras trabalhistas associadas às jornadas corridas. Todos os parâmetros usados são determinados por testes experimentais. Para garantir que todas as tarefas sejam cobertas por pelo menos uma jornada, jornadas com apenas uma tarefa são geradas e adicionadas ao *SP*.

4.2 Geração de Colunas

Como sugerido por Desrochers e Soumis (1989), problemas como este são do tipo não-determinístico polinomial difícil, ou *NP-hard*. O PPT pode ser formulado com um problema de particionamento de conjuntos. A formulação matemática se baseia no trabalho proposto por Lin e Hsu (2016), e é descrita da seguinte forma: seja D o conjunto de jornadas, que são geradas *a priori* partindo do *shift-pool*, e W o conjunto de tarefas.

Seja y_j uma variável de decisão binária indicando se a jornada j faz parte da solução ótima ou não. Além disso, seja $a_{ij} = 1$ se a tarefa i faz parte da jornada j e $a_{ij} = 0$ caso contrário. Define-se como c_j o custo da jornada j . O problema é então descrito como:

$$\text{Minimizar } \sum_{j \in D} c_j y_j \quad (4.1)$$

$$\text{s.t. } \sum_{j \in D} a_{ij} y_j = 1 \quad \forall i \in W \quad (4.2)$$

$$y_j \in \{0, 1\} \quad \forall j \in D \quad (4.3)$$

O programa garante que cada tarefa é atendida por um motorista e se torna claro que o número de colunas j para o problema pode ser razoavelmente grande. Assim, é impraticável gerar todas as colunas antecipadamente e, por isso, o algoritmo de geração de colunas deverá gerá-las conforme necessário, decompondo o problema em duas partes. A primeira parte é um problema de partição de conjuntos que garante que cada tarefa em um cronograma seja coberta por um conjunto de tarefas selecionado a partir de tarefas viáveis conhecidas. A segunda parte é o subproblema de *pricing* que gera tarefas viáveis para melhorar a solução incumbente do problema de partição de conjuntos da primeira parte.

Dada a dificuldade de enumerar todas as colunas viáveis e resolver as equações (4.1)-(4.3), deve-se primeiro construir um problema mestre restrito. Note que $j \in R$ no problema mestre restrito, ao invés de $j \in D$, onde R representa o subconjunto das jornadas viáveis de D , $R \subseteq D$.

4.2.1 Problema Mestre Restrito

$$\text{Minimizar } \sum_{j \in R} c_j y_j \quad (4.4)$$

$$\text{s.t. } \sum_{j \in R} a_{ij} y_j = 1 \quad \forall i \in W \quad (4.5)$$

$$y_j \in \{0, 1\} \quad \forall j \in R \quad (4.6)$$

4.2.2 Problema de Pricing

O problema dual das equações (4.1)-(4.3) pode ser formulado como a seguir:

$$\text{Maximizar } \sum_{i \in W} \pi_i \quad (4.7)$$

$$\text{s.t. } \sum_{i \in W} a_{ij} \pi_i \leq c_j \quad \forall j \in D \quad (4.8)$$

Onde π_i é a variável dual que corresponde a i -ésima restrição na equação (4.2). Denote c'_j como o custo reduzido da j -ésima coluna da formulação, como descrito em:

$$c'_j = c_j - \sum_{i \in W} a_{ij} \pi_i = \sum_{i \in W} a_{ij} c_j - \sum_{i \in W} a_{ij} \pi_i = \sum_{i \in W} a_{ij} (c_j - \pi_i) \quad (4.9)$$

Além disso, o modelo de *pricing* considera as seguintes variáveis:

Variáveis:

- d_j : 1 se a jornada j for do tipo dupla pegada, 0 caso contrário;
- e_j : quantidade de horas extras na jornada j ;
- o_j : quantidade de horas ociosas na jornada j ;
- E_{max} : limite máximo de horas extras permitidas por jornada;

Parâmetros:

- ρ : custo fixo do tripulante;
- θ : custo da dupla pegada;
- σ : custo da hora extra;
- ϕ : custo da hora ociosa.

$$\text{Minimizar } \sum_{j \in R} (\rho + \theta \cdot d_j + \sigma \cdot e_j + \phi \cdot o_j) \cdot y_j \quad (4.10)$$

$$\text{s.t. } e_j \leq E_{max} \quad \forall j \in R \quad (4.11)$$

$$d_j \in \{0, 1\} \quad \forall j \in R \quad (4.12)$$

A restrição (4.11) limita o número máximo de horas extras por jornada, com o valor de E_{max} definido como duas horas para este experimento, enquanto a restrição (4.12) impõe que a variável d_j seja binária, que indica se o custo da dupla pegada θ deve ser aplicado à jornada j ou não.

5 Resultados

Neste capítulo são apresentados os resultados dos experimentos computacionais obtidos pelo modelo de particionamento de conjuntos e pelo modelo de caminho mínimo com restrição de recursos. Os modelos foram implementados na linguagem de programação Python, com o framework Python-MIP (Santos *et al.*, 2020) e utilizando o solver CBC (COIN-OR Foundation, 2021). Todos os experimentos foram realizados utilizando o ambiente do Google Colab, com até 12 GB de RAM e um processador Intel(R) Xeon(R) CPU @ 2.20GHz de múltiplos núcleos. As instâncias utilizadas foram obtidas a partir do conjunto de dados da Bacia do Barreiro (Silva, 2005).

A organização deste capítulo é descrita a seguir. Inicialmente, na seção 5.1, são apresentados os testes realizados para o PPT com base no modelo proposto, utilizando dados reais de uma companhia de transporte público. Por fim, a seção 5.2 apresenta uma comparação entre a metodologia proposta e os resultados presentes na literatura.

5.1 Resultados para a Geração de Colunas partindo do Conjunto de Colunas Inicial Promissor

Para validar a metodologia proposta na resolução do Problema de Programação de Tripulação (PPT), um conjunto de empresas de transporte coletivo da região metropolitana de Belo Horizonte forneceu 32 instâncias do problema, representando situações reais de dias úteis, sábados e domingos na cidade. As segundas e sextas-feiras apresentam dados semelhantes, embora com algumas diferenças, atribuídas à natureza dos dias subsequentes.

Nos experimentos, foram utilizados os seguintes valores, comumente encontrados na literatura para os parâmetros correspondentes, conforme descrito em Souza (2014):

- Custo por jornada de trabalho: 10.000
- Custo da dupla pegada: 600 e 5.000
- Custo da hora extra: 240
- Custo da hora ociosa: 60

Outrossim, adotou-se uma tolerância de otimalidade de 10^{-6} , equilibrando eficiência computacional e precisão da solução, o que garante que o solucionador pare ao atingir uma solução dentro de 0,0001% do valor ótimo teórico, proporcionando alta precisão prática. Para

limitar o tempo computacional, definiu-se um tempo máximo de execução de 4 horas (14.400 segundos), assegurando uma solução em um período razoável.

Realizaram-se dois experimentos com o conjunto de entrada: no primeiro, atribuiu-se um custo de 600 para as jornadas do tipo *dupla pegada*, enquanto no segundo esse custo foi elevado para 5.000. Os demais parâmetros foram mantidos constantes entre os dois experimentos. Os resultados estão apresentados nas Tabelas 5.1 e 5.2, respectivamente.

As Tabelas 5.1 e 5.2 detalham, para cada instância, o nome (Problema), a quantidade de tarefas (Tamanho), e, em relação à solução, o número total de jornadas utilizadas (Jornadas), as horas extras (Horas Extras), o tempo ocioso total (Horas Ociosas), o número de jornadas do tipo dupla pegada (Dupla Pegada), o valor da função objetivo obtida (F.O.), e o tempo total necessário para encontrar a solução ótima, em segundos.

Tabela 5.1 – Resultados do PPT para a geração de colunas partindo do *shift-pool* com custo 600 para jornadas do tipo dupla pegada

Problema	Tamanho	Solução					Tempo Total (s)
		Jornadas	Horas Extras	Horas Ociosas	Dupla Pegada	F.O.	
DOM-G02	78	33	0,30	95,57	15	192.310	572,33
DOM-G27	37	15	0,07	31,23	6	106.455	77,46
DOM-G46	201	69	0,00	117,75	58	419.510	14.420,41
DOM-G48	235	128	0,00	626,65	17	351.629	10.360,99
DOM-G61	407	116	64,47	115,83	58	588.108	14.457,24
DOM-G69	249	49	0,00	350,35	49	482.806	14.432,27
DOM-G99	104	70	0,00	405,50	0	114.731	122,01
DOM-G115	81	28	3,30	26,20	24	239.404	675,23
SEG-G02	171	77	10,27	136,12	36	651.129	7.031,06
SEG-G27	73	27	4,52	25,27	18	225.406	503,68
SEG-G46	306	97	43,71	376,61	63	837.735	14.474,75
SEG-G48	300	127	0,00	456,72	56	613.870	14.430,40
SEG-G61	621	178	82,35	215,23	96	1.354.629	14.514,46
SEG-G69	387	106	31,86	117,24	68	1.078.957	14.452,21
SEG-G99	194	75	1,58	249,68	39	438.584	11.037,41
SEG-G115	139	59	18,42	53,62	34	531.645	2.077,46
SEX-G02	178	75	11,20	140,92	39	671.834	6.465,65
SEX-G27	71	24	5,22	20,15	16	248.000	375,33
SEX-G46	291	85	29,39	193,23	44	859.926	14.465,24
SEX-G48	321	88	57,17	137,04	46	627.807	14.452,12
SEX-G61	622	173	98,60	203,54	102	1.547.911	14.491,26
SEX-G69	375	104	65,42	132,37	86	1.204.458	14.509,00
SEX-G99	190	78	4,72	132,37	41	451.033	13.370,27
SEX-G115	138	56	26,02	40,78	27	527.176	2.234,23
SAB-G02	121	43	2,80	72,25	32	354.428	2.567,26
SAB-G27	55	19	1,27	34,52	16	145.456	510,94
SAB-G46	243	75	0,00	126,38	75	564.074	14.409,87
SAB-G48	269	64	44,90	38,64	41	441.918	14.487,54
SAB-G61	446	128	26,44	78,50	68	885.145	14.480,70
SAB-G69	302	66	19,30	87,92	46	648.819	14.427,50
SAB-G99	137	64	0,00	276,15	22	233.974	2.778,09
SAB-G115	100	33	6,17	27,48	28	286.083	1.076,20

Fonte: Elaborado pelo autor.

Tabela 5.2 – Resultados do PPT para a geração de colunas partindo do *shift-pool* com custo 5.000 para jornadas do tipo dupla pegada

Problema	Tamanho	Solução					Tempo Total (s)
		Jornadas	Horas Extras	Horas Ociosas	Dupla Pegada	F.O.	
DOM-G02	78	34	9,66	110,49	12	214.582	507,04
DOM-G27	37	16	14,23	43,63	4	204.309	65,89
DOM-G46	201	70	9,29	127,02	57	535.733	14.469,64
DOM-G48	235	129	11,3	638,21	15	494.949	9.144,11
DOM-G61	407	118	76,72	128,49	56	670.970	14.457,98
DOM-G69	249	50	13,97	356,11	46	610.289	14.408,01
DOM-G99	104	72	13,77	413,72	0	218.689	104,46
DOM-G115	81	29	13,88	32,16	20	345.188	604,06
SEG-G02	171	78	21,72	148,01	32	784.088	6.321,58
SEG-G27	73	28	16,35	38,45	14	270.092	446,10
SEG-G46	306	98	53,85	388,98	60	925.830	14.474,21
SEG-G48	300	128	12,47	470,69	55	642.823	14.430,05
SEG-G61	621	180	92,93	224,84	92	1.415.209	14.422,62
SEG-G69	387	107	39,67	123,8	66	1.204.568	14.414,54
SEG-G99	194	77	8,44	257,64	37	490.116	10.230,12
SEG-G115	139	60	24,98	58,76	32	680.174	1.895,43
SEX-G02	178	77	16,42	149,04	36	768.835	5.783,98
SEX-G27	71	26	12,47	26,25	15	374.455	349,49
SEX-G46	291	87	34,92	200,39	43	982.431	14.465,89
SEX-G48	321	90	64,18	142,81	42	788.263	14.423,13
SEX-G61	622	174	104,64	211,75	99	1.596.436	14.479,64
SEX-G69	375	106	70,72	139,92	85	1.262.388	14.509,73
SEX-G99	190	79	11,43	141,57	37	588.926	11.677,19
SEX-G115	138	57	31,03	51,59	26	551.710	2.049,79
SAB-G02	121	44	13,62	83,58	28	444.978	2.321,15
SAB-G27	55	21	9,49	46,81	13	196.791	469,88
SAB-G46	243	76	5,69	136,34	72	700.287	14.409,21
SAB-G48	269	66	52,58	44,25	39	535.166	14.487,00
SAB-G61	446	129	33,13	85,8	65	966.550	14.501,74
SAB-G69	302	68	29,73	96,67	44	734.847	14.427,74
SAB-G99	137	66	7,15	283,71	18	293.633	2.496,32
SAB-G115	100	34	17,94	36,86	27	442.734	933,92

Fonte: Elaborado pelo autor.

Convém destacar neste ponto que todos os descansos obrigatórios em jornadas superiores a quatro horas, sejam eles de 15 ou 30 minutos, foram contabilizados como tempo ocioso pelo algoritmo. Também vale notar que o limite de 4 horas foi estabelecido exclusivamente para a execução do algoritmo de geração de colunas, enquanto o tempo total exibido nas tabelas inclui qualquer pré-processamento adicional, incluindo a geração do *shift-pool*.

Os dados apresentados nas Tabelas 5.1 e 5.2 mostram que a redução do custo da dupla pegada promove uma configuração de jornadas mais flexível e balanceada para os motoristas. Esse ajuste reduz a necessidade de horas extras e, conseqüentemente, o desgaste dos profissionais, oferecendo uma solução mais equilibrada entre trabalho e descanso.

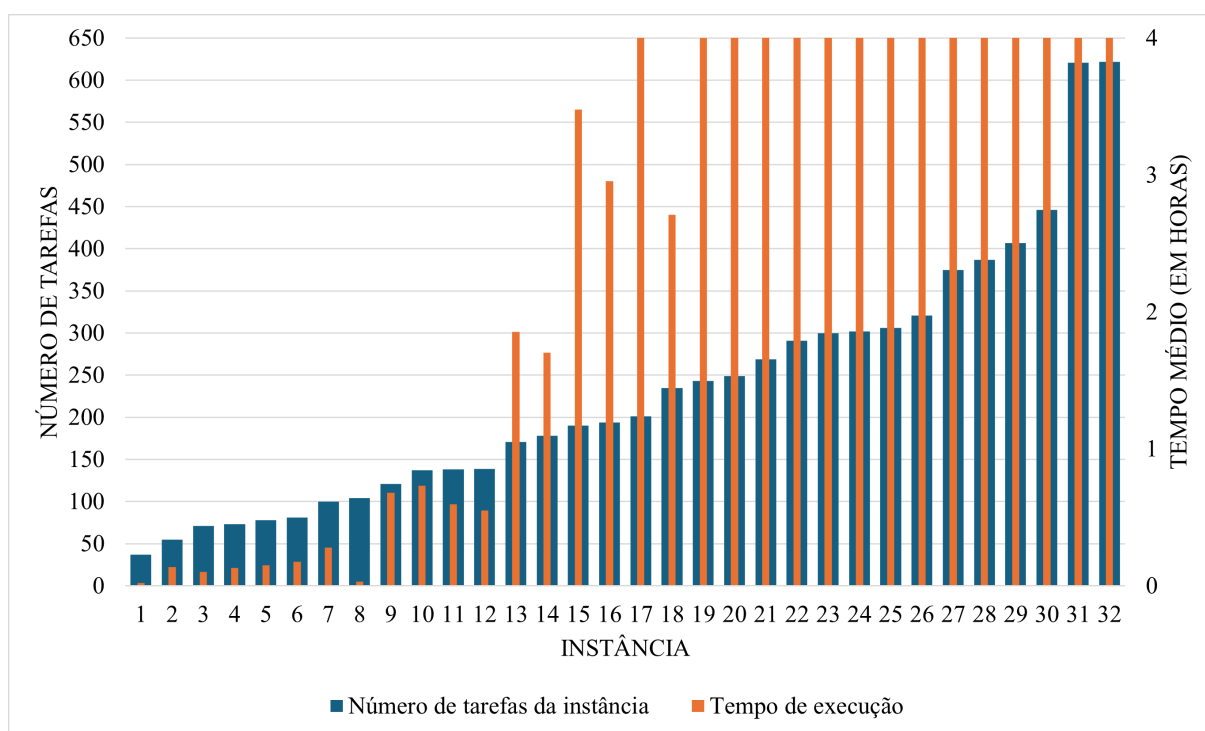
Por outro lado, ao elevar o custo da dupla pegada, nota-se um aumento nas horas ociosas e, em alguns casos, nas horas extras. Esse comportamento reflete um ajuste das jornadas para

evitar a dupla pegada, mas que pode comprometer a eficiência do uso do tempo disponível. O impacto dessa escolha é relevante tanto para a qualidade das condições de trabalho dos motoristas quanto para a eficiência operacional da empresa, permitindo decisões estratégicas sobre o custo da dupla pegada conforme a demanda semanal. Essa comparação entre os cenários será detalhada na Seção 5.2, onde a Tabela 5.4 resume as variações observadas entre os custos de dupla pegada.

Além disso, os resultados de ambas as tabelas demonstram que o algoritmo obteve soluções ótimas em 17 das 32 instâncias, dentro do limite de 4 horas. Essas instâncias estão destacadas em negrito nas tabelas. Para as instâncias restantes, o *solver* produziu soluções factíveis ao alcançar o limite de 4 horas.

O gráfico da figura 5.1 apresenta o eixo X, que indica o número da instância, e dois eixos Y. O eixo Y à esquerda representa o número de tarefas em cada instância, enquanto o eixo Y à direita indica o tempo médio, em horas, necessário para produzir uma solução ótima ou interromper a execução com uma solução factível. As instâncias estão organizadas em ordem crescente, de acordo com o número de tarefas que contêm.

Figura 5.1 – Tempo médio para resolver x quantidade de tarefas



Fonte: Elaborado pelo autor.

A figura 5.1 também indica que, para instâncias com cerca de 250 tarefas ou mais, o tempo necessário para o algoritmo finalizar tende a crescer, frequentemente ultrapassando o limite de 4 horas. Em contrapartida, para instâncias menores, a execução é substancialmente mais rápida, com tempos que, em muitos casos, são inferiores a uma hora.

5.2 Análise Comparativa dos Resultados

Houve interesse em realizar um comparativo direto com os trabalhos de Souza (2014) e Cunha (2021). Para tanto, foram conduzidas duas experimentações empregando os mesmos parâmetros utilizados por Souza (2014), a fim de verificar o desempenho dos métodos propostos em condições similares. No entanto, devido à dificuldade de obter dados de entrada exatamente iguais aos utilizados pelos autores mencionados, a comparação busca destacar as diferenças qualitativas nas abordagens e nos resultados observados. Dessa forma, a análise é orientada pela avaliação dos parâmetros e das estratégias aplicadas em cada estudo, respeitando as limitações dos dados disponíveis.

A análise dos resultados para o domingo, representada na Tabela 5.3, foi escolhida de forma arbitrária, mas fornece uma amostra diversificada no tamanho das instâncias e representativa do conjunto completo, permitindo uma análise comparativa dos parâmetros avaliados em cada um dos experimentos.

Tabela 5.3 – Resultados Seleccionados dos Experimentos para Viagens de Ônibus aos Domingos

Problema	Custo Dupla Pegada	Jornadas	Horas Extras	Horas Ociosas	Dupla Pegada	F.O.
DOM-G02	600	33	0,3	95,57	15	192.310
	5.000	34	9,66	110,49	12	214.582
DOM-G27	600	15	0,07	31,23	6	106.455
	5.000	16	14,23	43,63	4	204.309
DOM-G46	600	69	0,0	117,75	58	419.510
	5.000	70	9,29	127,02	57	535.733
DOM-G48	600	128	0,0	626,65	17	351.629
	5.000	129	11,3	638,21	15	494.949
DOM-G61	600	116	64,47	115,83	58	588.108
	5.000	118	76,72	128,49	56	670.970
DOM-G69	600	49	0,0	350,35	49	482.806
	5.000	50	13,97	356,11	46	610.289
DOM-G99	600	70	0,0	405,5	0	114.731
	5.000	72	13,77	413,72	0	218.689
DOM-G115	600	28	3,3	26,2	24	239.404
	5.000	29	13,88	32,16	20	345.188

Fonte: Elaborado pelo autor.

A fim de analisar o impacto dos diferentes custos atribuídos à dupla pegada, foi calculada a variação percentual dos resultados entre os cenários com custos de 600 e 5.000. Esse cálculo foi realizado utilizando a fórmula:

$$\frac{\text{resultado com custo 600} - \text{resultado com custo 5.000}}{\text{resultado com custo 5.000}} \times 100$$

A expressão quantifica a diferença percentual entre os resultados obtidos nos dois cenários, tomando o valor do cenário com custo 5.000 como referência. Assim, é possível observar a variação dos parâmetros — função objetivo, horas extras, horas ociosas e número de jornadas —

em resposta à mudança nos custos de dupla pegada. Esse comparativo permite avaliar o desvio percentual dos resultados para o custo 600 em relação aos obtidos para o custo mais elevado, fornecendo uma visão detalhada sobre o efeito dessa variável nos resultados finais, dispostos na Tabela 5.4.

Tabela 5.4 – Variação Percentual dos Resultados entre Custos de Dupla Pegada (600 e 5.000)

	F.O.	Total de Jornadas	Total de Horas Extras	Total de Horas Ociosas	Total de Duplas Pegadas
DOM-G02	-10,38%	-2,94%	-96,89%	-13,50%	25,00%
DOM-G27	-47,90%	-6,25%	-99,51%	-28,42%	50,00%
DOM-G46	-21,69%	-1,43%	-100,00%	-7,30%	1,75%
DOM-G48	-28,96%	-0,78%	-100,00%	-1,81%	13,33%
DOM-G61	-12,35%	-1,69%	-15,97%	-9,85%	3,57%
DOM-G69	-20,89%	-2,00%	-100,00%	-1,62%	6,52%
DOM-G99	-47,54%	-2,78%	-100,00%	-1,99%	0,00%
DOM-G115	-30,65%	-3,45%	-76,22%	-18,53%	20,00%

Fonte: Elaborado pelo autor.

Observa-se, a partir desses resultados, que a variação percentual entre os custos de dupla pegada (600 e 5.000) configura um impacto significativo sobre diversas métricas analisadas. A redução na função objetivo é evidente em todos os casos, indicando que um custo menor para dupla pegada possibilita uma otimização mais eficiente nos valores desta. Além disso, o número total de jornadas apresenta uma leve diminuição, embora a variação seja menos expressiva em comparação com outras métricas.

A redução nas horas extras é substancial, chegando a valores próximos de 100% para a maioria dos problemas, o que sugere que um custo menor para dupla pegada desestimula o uso desse recurso. Esse ajuste também reflete no aumento percentual de horas ociosas e no número de duplas pegadas, mostrando uma tendência de maior ociosidade e maior utilização de jornadas de dupla pegada quando o custo é mais baixo.

Esses resultados indicam que a configuração de custo para dupla pegada influencia diretamente a estratégia de alocação de jornadas. O custo menor promove maior flexibilidade no uso de duplas pegadas e uma distribuição mais equilibrada entre horas extras e horas ociosas, adaptando-se às demandas de forma a otimizar o valor da função objetivo e reduzir o custo total das jornadas.

Embora o presente estudo não tenha realizado uma comparação direta com o trabalho de Souza (2014), os tempos de processamento observados indicam uma eficiência computacional superior. A proposta do autor visa minimizar o tempo de cálculo, evitando a geração antecipada ou intermediária de jornadas, o que implica na construção das jornadas somente à medida que o processo de otimização avança. No entanto, ele reporta dificuldades em calcular soluções para problemas com mais de 50 jornadas geradas. Em contraste, o método aqui proposto inclui um conjunto viável de colunas desde o início do processo de otimização, permitindo uma exploração mais ampla das soluções possíveis. Esse procedimento não encontrou a mesma

limitação, sendo capaz de gerar uma solução para a instância DOM-G48 com um total de 128 jornadas, demonstrando uma capacidade significativamente ampliada de lidar com instâncias de maior complexidade sem impacto crítico no tempo de processamento.

6 Conclusão e Trabalhos Futuros

Neste trabalho, foi implementado um modelo para o Problema de Programação de Tripulação (PPT) utilizando a técnica de geração de colunas, considerando o cenário de empresas de transporte público da região metropolitana de Belo Horizonte. O modelo desenvolvido utilizou um *shift-pool* inicial promissor que, aliado ao algoritmo de geração de colunas, permitiu resolver diversas instâncias de forma eficiente, mesmo em problemas de maior complexidade.

A abordagem adotada possibilitou a obtenção de soluções próximas à otimalidade, com limites inferiores de alta qualidade, especialmente nas instâncias menores e de média complexidade. Para as instâncias maiores, o algoritmo demonstrou robustez ao produzir soluções factíveis dentro do tempo limite de execução, mostrando a relevância do conjunto inicial de colunas promissoras. Comparativamente, os resultados obtidos foram competitivos com métodos exatos da literatura, alcançando desempenho similar em menor tempo de processamento.

Entre as contribuições deste trabalho, destaca-se a seleção eficiente de jornadas viáveis que atendem às restrições legais e operacionais, promovendo uma função objetivo otimizada. Além disso, foi adotada uma tolerância de otimalidade que garante que o *solver* interrompa o processo próximo ao valor ótimo, equilibrando precisão prática e eficiência computacional. Essa abordagem aprimora a alocação de motoristas, reduzindo o número de jornadas necessárias e gerando escalas de trabalho mais econômicas e ajustadas às exigências do transporte público. Em comparação com abordagens anteriores, o método proposto neste estudo, ao incluir um conjunto viável de colunas desde o início do processo de otimização, permite uma exploração mais abrangente das soluções possíveis. Isso resultou em uma capacidade significativamente ampliada de lidar com instâncias de maior complexidade, sem impacto crítico no tempo de processamento, superando limitações observadas em métodos que geram jornadas apenas conforme o processo avança.

Como trabalhos futuros, recomenda-se: (i) a aplicação de heurísticas ou matheurísticas, como ALNS ou GRASP, diretamente no *shift-pool*, visando selecionar jornadas promissoras de forma adaptativa e iterativa para melhorar tanto o tempo de execução quanto a qualidade das soluções; (ii) o desenvolvimento de critérios de priorização no *shift-pool*, permitindo a classificação das jornadas pela eficiência potencial, o que reduziria o tempo de convergência; e (iii) a exploração de ajustes dinâmicos nos parâmetros de controle do algoritmo, como a frequência de adição de novas jornadas, permitindo ao modelo focar em regiões de alta qualidade de solução sem comprometer a precisão.

Referências

CHEN, S.; SHEN, Y. An improved column generation algorithm for crew scheduling problems. **Journal of Information and Computational Science**, v. 10, n. 1, p. 175–183, 2013.

COIN-OR Foundation. **CBC (Coin-or branch and cut)**. 2021. Disponível em: <<https://www.coin-or.org/Cbc>>. Acesso em: 25 set. 2024.

CONSTANTINO, A. A. *et al.* Solving a large real-world bus driver scheduling problem with a multi-assignment based heuristic algorithm. Graz Univ Technolgy, Inst Information Systems Computer Media-iicm, 2017.

CUNHA, J. d. S. **Modelos e heurísticas para os problemas de programação de veículos e de tripulações no sistema de transporte público**. 41 p. Monografia (Graduação em Ciência da Computação) — Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Ouro Preto, 2021.

DANTZIG, G. B. **Linear Programming and Extensions**. Princeton: Princeton University Press, 1963. 625 p. ISBN 9780691080000.

DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. **Operations research, INFORMS**, v. 8, n. 1, p. 101–111, 1960.

DESROCHERS, M.; SOUMIS, F. A column generation approach to the urban transit crew scheduling problem. **Transportation science, INFORMS**, v. 23, n. 1, p. 1–13, 1989.

DESROSIERS, J.; LUBBECKE, M. A primer in column generation. In: DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M. (Ed.). **Column Generation**. Boston, MA: Springer, 2006. p. 1–32. ISBN 0-387-25485-4.

FISCHETTI, M.; MARTELLO, S.; TOTH, P. The fixed job schedule problem with spread-time constraints. **Operations Research, INFORMS**, v. 35, n. 6, p. 849–858, 1987.

FISCHETTI, M.; MARTELLO, S.; TOTH, P. The fixed job schedule problem with working-time constraints. **Operations Research, INFORMS**, v. 37, n. 3, p. 395–403, 1989.

HUISMAN, D. **Integrated and Dynamic Vehicle and Crew Scheduling**. Tese (Doutorado) — Thela Thesis, Tinbergen Instituut Research Series, Amsterdam, 2004.

IBARRA-ROJAS, O. J. *et al.* Planning, operation, and control of bus transport systems: A literature review. **Transportation Research Part B: Methodological**, Elsevier, v. 77, p. 38–75, 2015.

JALON, M. L. D.; LARA, M. A. Sanchez de; GELASHVILI, V. Explanatory factors for public transportation financing needs in Spain. **Journal of Advanced Transportation**, Hindawi, v. 2019, 2019.

LIN, D.-Y.; HSU, C.-L. A column generation algorithm for the bus driver scheduling problem. **Journal of Advanced Transportation**, Wiley Online Library, v. 50, n. 8, p. 1598–1615, 2016.

- MARTINS, L. d. C. **O uso da heurística Adaptive Large Neighborhood Search para resolver o Problema de Programação de Tripulações do Transporte Público**. Dissertação (Mestrado) — Instituto de Ciências Exatas e Biológicas, Universidade Federal de Ouro Preto, Ouro Preto, 2017.
- PERUMAL, S. **Integrated optimization of vehicle and crew scheduling in public transport**. Tese (Doutorado) — PhD thesis, Technical University of Denmark, Kongens Lyngby, 2004.
- PERUMAL, S. S. *et al.* A matheuristic for the driver scheduling problem with staff cars. **European Journal of Operational Research**, Elsevier, v. 275, n. 1, p. 280–294, 2019.
- SANTOS, H. G. *et al.* **coin-or/python-mip: 1.11.0**. 2020. Disponível em: <<https://www.github.com/coin-or/python-mip>>. Acesso em: 25 set. 2024.
- SILVA, G. P. **Bacia do Barreiro - Conjunto de Instâncias**. 2005. Acesso em: 14 nov. 2024. Disponível em: <http://www.decom.ufop.br/gustavo/Bacia_do_Barreiro/Bacia_do_Barreiro.htm>.
- SILVA, G. P.; CUNHA, C. B. da. Uso da técnica de busca em vizinhança de grande porte para a programação da escala de motoristas de ônibus urbano. **Transportes**, v. 18, n. 2, 2010.
- SILVA, G. P.; REIS, A. F. d. S. A study of different metaheuristics to solve the urban transit crew scheduling problem. **Journal of Transport Literature**, SciELO Brasil, v. 8, p. 227–251, 2014.
- SILVA, T. A.; SILVA, G. P. O uso da metaheurística guided local search para resolver o problema de escala de motoristas de ônibus urbano. **Transportes**, v. 23, n. 2, p. 105–116, 2015.
- SOUZA, D. S. **Uma abordagem híbrida para resolver o problema da escala de motoristas de ônibus urbano**. 83 p. Dissertação (Mestrado) — Universidade Federal de Ouro Preto, Ouro Preto, 2014.
- VALDES, V. A. V. **Integrating Crew Scheduling and Rostering Problems**. Tese (Doutorado) — Alma Mater Studiorum Università di Bologna, Bologna, 2010. Dottorato di ricerca in Automatica e ricerca operativa, 21 Ciclo.
- WINSTON, W. L. **Operations Research: Applications and Algorithms**. 4th. ed. Australia; Belmont, CA: Thomson Brooks/Cole, 2004. 1418 p. ISBN 9780534423629.