

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO DE JOINVILLE  
ENGENHARIA DE TRANSPORTES E LOGÍSTICA

ISADORA DE ALBUQUERQUE RODRIGUES

UM MÉTODO HEURÍSTICO PARA O PROBLEMA DE ALOCAÇÃO DE TRIPULAÇÃO  
DE ÔNIBUS URBANO

Joinville  
2016

ISADORA DE ALBUQUERQUE RODRIGUES

UM MÉTODO HEURÍSTICO PARA O PROBLEMA DE ALOCAÇÃO DE TRIPULAÇÃO  
DE ÔNIBUS URBANO

Trabalho de Conclusão de Curso submetido à  
Universidade Federal de Santa Catarina como  
parte dos requisitos necessários para a obtenção  
do Grau de Bacharel em Engenharia de  
Transportes e Logística.

Orientadora: Dra. Silvia Lopes da Sena  
Tagliarenha.

Coorientador: Dr. Pablo Andretta Jaskowiak.

Joinville  
2016

## **AGRADECIMENTOS**

Aos meus pais, meus maiores exemplos de tudo, meu porto seguro e sempre me incentivando a ir a mais do que eu achava que era capaz.

À Silvia e ao Pablo, que foram além de professores bons orientadores, colaborando de forma fundamental ao trabalho e a minha formação.

Aos professores deste curso, especialmente a Elisete, Janaina e Christiane, que são exemplos, apoiando e motivando a mais esta jornada.

Ao Diego e ao Kaique, que me incentivaram a aprofundar meus conhecimentos neste trabalho.

À Ghislaine por ter sido além de colega, Amiga.

Aos meus amigos presentes em minha vida.

## RESUMO

Neste trabalho abordou-se o problema de programação de tripulantes (PPT) de ônibus urbano, o qual consiste em atribuir a cada tripulação uma sequência de viagens. Como essas viagens possuem intervalos de duração diferentes, cada tripulação poderá executar viagens com acúmulo de horas extras ou ociosas. Propôs-se um modelo heurístico para a geração de escalas de motoristas de forma a otimizar a alocação de tripulações de ônibus urbano, baseado em caminhos mínimos e atendendo as restrições operacionais e trabalhistas. A redução dos custos na escala de trabalho como a diminuição de trabalhadores ou de horas pagas, afeta tanto as empresas operadoras como os usuários do serviço, pois há a possibilidade de investimento na melhoria da qualidade do transporte público ou na redução do preço da tarifa. Para a validação do método proposto foram utilizados dados de uma empresa de transporte público de médio porte na cidade de Joinville, considerando-se apenas as escalas para dias úteis. O método proposto mostrou-se eficiente para a obtenção de diversas soluções iniciais para aplicar outros métodos heurísticos de busca local de otimização.

**Palavras-chave:** Problema de programação de tripulação, Caminhos mínimos, Heurísticas.

## **ABSTRACT**

In this work the problem of crew scheduling (PPT) of urban buses was discussed, which consists in assigning to each crew a sequence of trips. Because these trips have different lengths of time, each crew can travel with overtime or idle accumulation. It was proposed a heuristic model for the generation of driver scales in order to optimize the allocation of urban bus crews, based on minimum paths and taking into account the operational and labor restrictions. Reducing work-scale costs such as the reduction of workers or hours paid affects both operating companies and users of the service as there is the possibility of investment in improving the quality of public transport or reducing the price of the tariff. To validate the proposed method, we used data from a medium-sized public transport company in the city of Joinville, considering only the scales for business days. The proposed method proved to be efficient to obtain several initial solutions to apply other heuristic local search optimization methods.

**Keywords:** Crew scheduling problem, Minimum paths, Heuristics.

## LISTA DE SIGLAS

IPPUJ	Instituto de Pesquisa e Planejamento Urbano de Joinville
PO	Pesquisa operacional
PL	Programação linear
PNL	Programação não linear
PPT	Problema de programação de tripulação
PLI	Programação linear inteira
VNS	Variable neighborhood search
VND	Variable neighborhood descent
VLNS	Large variable neighborhood search
TS	Tabu search

## Sumário

1	INTRODUÇÃO .....	8
1.1	JUSTIFICATIVA .....	10
1.2	OBJETIVOS .....	11
1.3	METODOLOGIA .....	12
1.4	ESTRUTURA DO TRABALHO .....	13
2	FUNDAMENTAÇÃO TEÓRICA .....	14
2.1	O PROBLEMA DE PROGRAMAÇÃO DE TRIPULAÇÕES .....	15
2.2	MODELAGEM MATEMÁTICA .....	17
2.3	MÉTODOS DE SOLUÇÃO .....	20
2.3.1	MÉTODOS EXATOS .....	21
2.3.2	MÉTODOS APROXIMADOS .....	27
3	MÉTODO PROPOSTO .....	33
3.1	ESCALA ATUAL DA EMPRESA .....	33
3.2	ABORDAGEM PROPOSTA.....	35
3.3	PARÂMETROS .....	37
3.4	IMPLEMENTAÇÃO.....	38
3.5	RESULTADOS OBTIDOS .....	42
4	CONSIDERAÇÕES FINAIS .....	50
	REFERÊNCIAS.....	52

## 1 INTRODUÇÃO

A cidade de Joinville em Santa Catarina tem como principal forma de transporte público o ônibus, com um sistema de tarifa única de bilhetagem automática com integração temporal. São ao todo 10 estações e terminais, a fim de operar com integração física e temporal, com um desenho tronco-alimentador e com forma radial e diametral. Ao todo são 257 linhas regulares, compostas por 21 linhas troncais entre paradoras e diretas, 17 linhas de interestações, 184 linhas alimentadoras especiais e 4 linhas de vizinhança, informado pela prefeitura de acordo com Joinville Bairro a Bairro (2015).

De acordo com a pesquisa do IPPUJ de 2015, em parceria com as empresas de transporte coletivo da cidade de Joinville, Transtusa e Gidion, a média de passageiros transportados por dia em comparação ao ano de 2014 com o ano de 2000 reduziu em quase 19 mil pessoas, ao mesmo tempo que houve um crescimento populacional de 1,69% conforme detalhes da Tabela 1. Um dos motivos para esta queda foi o aumento na tarifa. Segundo o censo 2010, feito pelo IBGE, 19,6% da receita familiar é utilizada para o transporte (ROCHA, 2012).

Tabela 1: Indicadores de passageiros transportados em Joinville.

<b>Ano</b>	<b>Passageiros transportados dia (média)</b>	<b>População</b>	<b>Indicador (passageiros transportados por dia / população)</b>
2000	139.022	429.604	32,36 %
2010	128.106	515.288	24,86 %
2011	130.467	520.905	25,04 %
2012	127.415	526.338	24,20 %
2013	121.726	546.981	22,25 %
2014	120.040	554.601	21,64 %

Fonte: (IPPUJ, 2015).

O Instituto de Pesquisa e Planejamento Urbano de Joinville (IPPUJ) junto à empresa EMBARQ Brasil, aplicaram o programa de qualidade do serviço de ônibus, a QualiÔnibus em novembro de 2014. Os parâmetros com melhor avaliação pelos usuários foram a



facilidade de pagamento e o atendimento ao cliente. Os piores parâmetros apontados na pesquisa foram o custo da passagem e o conforto dos pontos de ônibus. Os usuários apresentaram em média uma satisfação ao transporte oferecido de 5,4 em uma escala de 0 a 10 (QualiÔnibus, 2014). Em relação aos parâmetros qualitativos da pesquisa, os usuários mantêm uma posição indiferente. Estes parâmetros podem mudar, havendo uma redução de custo na empresa, o que pode impactar em tarifas menores e/ou aumento da qualidade do serviço.

Com o intuito de atrair mais usuários o setor do transporte público pode-se verificar a possibilidade de redução do valor da tarifa. Uma medida notável para atender essa expectativa, seria a redução do número de motoristas, ou seja, é necessário que haja uma reorganização de todas as escalas de motoristas visando uma minimização de custo. Prever um itinerário para cada motorista implicam algumas restrições, dentre as quais são principais: a carga horária de trabalho não deve ultrapassar os limites da lei; o tempo de descanso entre as jornadas; alocação de horário de trabalho de maneira econômica para empresa; existência de férias e atestados; folgas em finais de semana; horário diferenciado em feriados e domingos e a existência de outros terminais de transporte. Estes fatores revelam a importância de estudos matemáticos que visam bons resultados na otimização das escalas.

Segundo Silva e Cunha (2010) em empresas de transporte público, a utilização de ferramentas computacionais para dimensionar a frota e a tripulação de forma econômica é pequena. Por haver uma série de restrições ao modelo, segundo Hillier e Lieberman (2013) não há algoritmos que consigam obter a solução ótima em tempo polinomial da entrada de dados. Sendo assim, o desenvolvimento de métodos heurísticos para a obtenção de resultados de forma satisfatória para os problemas de grande porte traz grandes contribuições acadêmicas.

Uma vez desenvolvidos métodos eficientes que possibilitem a otimização da alocação de tripulação, o prestador de serviço terá, possivelmente, um aumento em sua receita final, e, portanto, poderá investir na qualidade do serviço, o que acarretará em um maior equilíbrio de interesses entre o operador e o usuário, ou seja, o problema também tem grande relevância econômica e operacional.

A empresa deste estudo de caso é de médio porte, possui cerca de 850 funcionários, uma frota de 260 veículos com mais de 4500 horários de ônibus distribuídos em 116 itinerários. A sugestão de uma nova programação de escalas foi realizada levando em conta o terminal central da cidade.

As escalas da empresa são feitas atualmente pelo setor de operações logísticas, em que

os profissionais da área atuam há bastante tempo, apresentando conhecimento suficiente da maior parte dos motoristas e veículos. Porém essa metodologia quando aplicada para uma larga escala de dados pode apresentar soluções distantes da ótima. Assim, o tempo ocioso entre as viagens realizadas torna-se crescente, gerando custos maiores à empresa em função de horas extras ou contratação de mais motoristas.

## 1.1 JUSTIFICATIVA

O problema da programação de tripulações – PPT (crew scheduling problem), é classificado como um problema NP-difícil (SOUZA, 2003) e consiste em distribuir tarefas aos tripulantes (pilotos, maquinistas, motoristas, aeromoças, cobradores, etc.) durante vários trechos de viagens a serem cobertos por empresas aéreas, ferroviárias ou rodoviárias, ou mesmo designar atividades pré-definidas (treinamentos, exames médicos e folgas, entre outros), gerando escalas (normalmente mensais) individuais para cada tripulante.

Conforme Fischetti et al (1987) o problema de programação de tripulações pode ser modelado de diferentes maneiras, mas é sempre classificado como um problema NP-difícil, já que os algoritmos existentes exigem um alto custo computacional para obter a solução ótima (SOUZA, 2014). Sendo assim, ocorre a necessidade da utilização de métodos heurísticos para resolver o problema de forma satisfatória, ou seja, de forma a determinar uma razoável solução em um tempo computacional viável.

Os problemas que são classificados como NP-difícil, são problemas com algoritmos baseados em enumeração, quer seja ela implícita ou não. O número de combinações possíveis é grande, fazendo com que os algoritmos enumerativos não consigam resolver problemas com um grande número de entradas em tempo hábil. Pode haver duas classificações para o problema: NP-completo e NP-difícil. Os NP-completos possuem uma forte evidência da não existência de um algoritmo cujo tempo de solução seja uma função polinomial do tamanho de entrada. Já os problemas classificados como NP-difícil são aqueles cujos problemas de decisão correspondente são NP-completos (COLIN, 2015).

A distribuição das tarefas deve ser feita de forma a otimizar a escala de cada tripulante, minimizando assim os gastos com tripulações. Devem ser respeitadas ainda as legislações vigentes, acordos sindicais e normas das empresas, que regulam a relação entre

empresas de transporte e tripulantes, de tal forma que todas as rotas oferecidas sejam cobertas pelos tripulantes.

O PPT vem sendo estudado desde a década de 60 tendo Elias (1964) como o pioneiro na área (SOUZA, 2014). Devido a mudanças no sistema de transporte público, diversas pesquisas são realizadas para tentar representar e resolver o problema da forma mais econômica e eficiente. Nas empresas de transporte público, a utilização de ferramentas computacionais para dimensionar suas frotas ainda é pequena, assim como para alocar seus motoristas e cobradores de maneira econômica e equilibrada (SILVA, 2001).

Desta forma, surge a importância de estudar e aplicar métodos de otimização nesta fase do processo de planejamento das empresas de transporte público para permitir a redução dos seus custos operacionais. Alguns modelos matemáticos conseguem auxiliar neste sequenciamento de tarefas, porém para cada empresa é necessário um modelo de programação diferente, visto que algumas apresentam restrições específicas não relacionadas as leis trabalhistas.

## **1.2 OBJETIVOS**

O objetivo geral deste trabalho é propor um modelo heurístico para a geração de escalas de motoristas de forma a otimizar a alocação de tripulações de ônibus urbano.

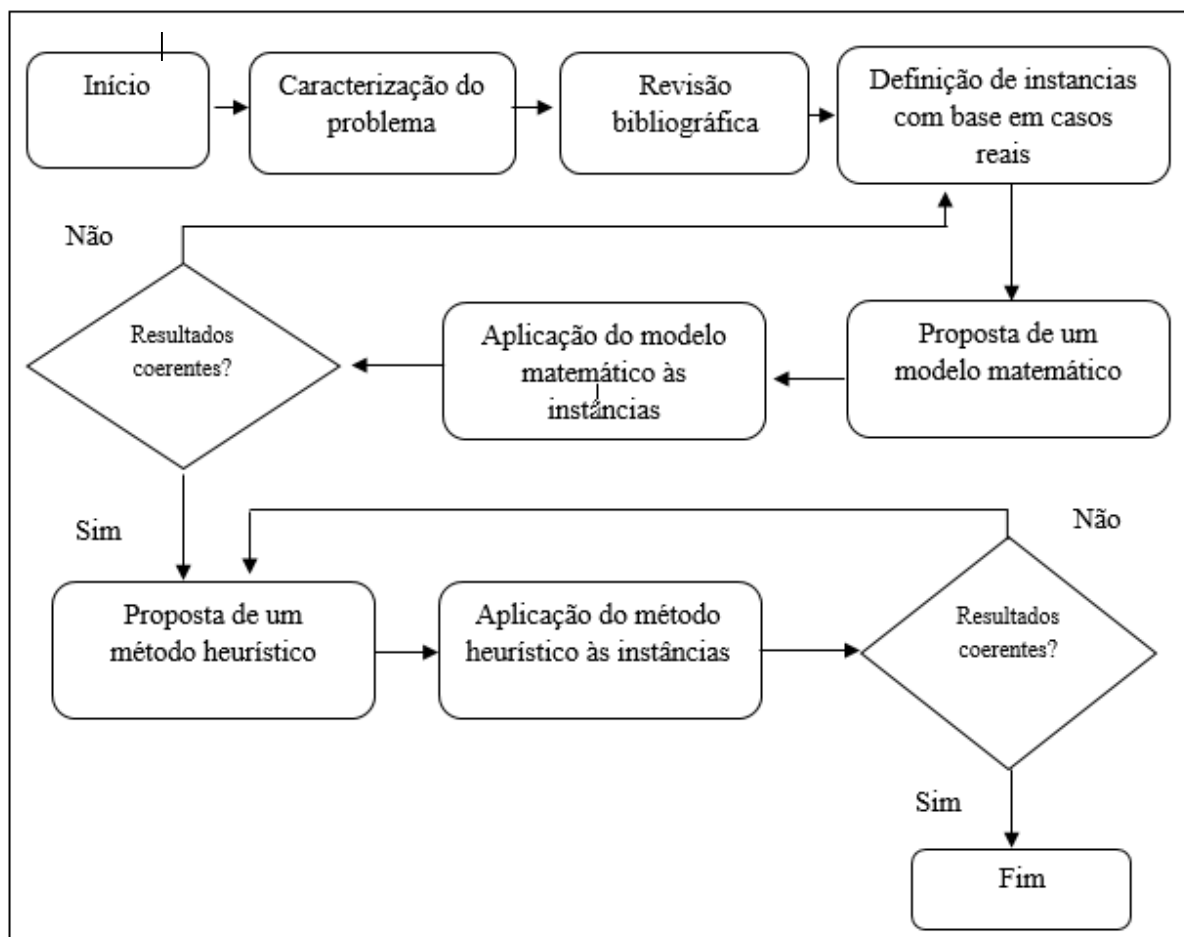
Para que o objetivo proposto seja alcançado, durante o desenvolvimento do trabalho, será necessário alcançar os seguintes objetivos específicos:

- Realizar levantamento bibliográfico;
- Levantar dados em uma empresa de transporte público;
- Apresentar um modelo matemático para otimizar o PPT;
- Determinar as restrições para o modelo matemático a ser usado na empresa em questão;
- Otimizar as escalas das tripulações, minimizando o tempo ocioso entre as tarefas;
- Propor um método de solução, neste caso, baseado em caminhos mínimos.

### 1.3 METODOLOGIA

Primeiramente será caracterizado o problema de programação de tripulação, junto as referências bibliográficas, mostrando diferentes métodos utilizados afim de se encontrar a solução do problema. Pela definição das instâncias com base em casos reais é proposto e aplicado um modelo matemático ao mesmo. Com resultados coerentes da etapa anterior se propõem e aplica-se um método heurístico as instancias até a obtenção de resultados coerentes. Esta é a metodologia seguida e apresentada na Figura 1.

Figura 1: metodologia do trabalho.



Fonte: Elaboração própria (2016).

## **1.4 ESTRUTURA DO TRABALHO**

O presente trabalho está estruturado em quatro capítulos, incluindo este introdutório. No segundo capítulo formaliza-se a definição do PPT e apresenta-se uma revisão teórica sobre o tema. Em seguida, são descritos, alguns métodos exatos e aproximados para a resolução do problema de programação de veículos relacionados a trabalhos semelhantes na literatura. No terceiro, apresenta-se a metodologia utilizada na realização deste estudo, as etapas do método proposto para a obtenção das novas escalas dos motoristas, assim como a solução inicial, a avaliação das escalas fornecidas, os parâmetros utilizados na implementação e finaliza-se com os resultados obtidos. O último capítulo apresenta as conclusões e algumas sugestões para trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

A pesquisa operacional, também conhecida como PO, surgiu em razão da necessidade de alocar, de forma eficiente, os escassos recursos nas diferentes operações militares na segunda guerra mundial. Cientistas do mundo todo, no período entre 1939 e 1945, foram convocados para solucionar problemas estratégicos e táticos de forma matemática, para fundamentar as tomadas de decisões de ações militares (HILLIER; LIEBERMAN, 2013).

Ao final da guerra a PO não ficaria de lado, visto que, era crescente o número de profissionais especializados na área. Setores comerciais, indústria e governo utilizavam a PO, o que levou a uma rápida disseminação de toda a informação para diversos países. O impacto foi impressionante para melhorias das organizações mundiais (HILLIER; LIEBERMAN, 2013).

A medida que aumenta a complexidade e a especialização de algum problema, torna-se mais difícil conseguir alocar de forma eficiente os recursos disponíveis a cada atividade do mesmo. Em razão desta dificuldade, e da necessidade de realizar cada tarefa da melhor forma, torna-se evidente a importância do uso da PO. Uma característica da pesquisa operacional é a utilização de modelos, o que facilita muito o processo de análise. Isto permite que uma decisão possa ser simulada antes de ser efetivamente implementada, o que gera economia de recursos e experiência, justificando assim o uso da PO como instrumento de gerência (ANDRADE, 2000).

Os métodos exatos ou analíticos, geralmente utilizam técnicas de decomposição matemática e técnicas de otimização clássica como: programação linear (PL), programação dinâmica, programação não linear (PNL) e programação inteira mista. Em geral, apresentam a característica de determinar a solução ótima do problema, são muito eficientes em problemas de pequeno e médio porte, mas para os de grande porte ainda apresentam problemas de convergência e de elevado esforço computacional.

Os métodos aproximados, também conhecidos como métodos heurísticos são as atuais alternativas para os modelos de otimização matemática. O termo heurística caracteriza-se por

ser um método de resolver problemas através de técnicas práticas aprendidas em experiências passadas; formado da palavra grega “heuriskein”, que significa descobrir, inventar, ter uma ideia, é utilizado para descrever todas essas técnicas que, ao invés de usar a otimização clássica, gera, passo a passo, uma solução avaliando e selecionando opções, com ou sem a ajuda do planejador. Para isto, nos modelos heurísticos buscas locais são realizadas com a orientação lógica ou empírica de índices de sensibilidade (regras heurísticas). Estas regras são utilizadas para gerar e classificar as opções de construção de uma solução de boa qualidade durante a busca. Em geral, estes métodos apresentam a vantagem de fornecer soluções de boa qualidade com esforços computacionais moderados, mas também não garantem que se encontre a solução ótima de problemas reais e não fornecem informações sobre a qualidade da solução obtida. Entretanto, estes métodos ainda são os mais utilizados para problemas de alta complexidade (TAGLIALENHA, 2008).

Um problema de alta complexidade bastante relevante nos estudos de PO é a alocação de força de trabalho e a programação de trabalhadores de modo a atender à demanda por recursos que variam ao longo do tempo, como por exemplo, alocação de operadores em telefonia, enfermeiras em hospitais, policiais e pessoas de transporte (tripulação de avião, motoristas de ônibus, etc.). Dentre estes problemas, destaca-se o problema da programação de tripulações, o qual será abordado neste capítulo.

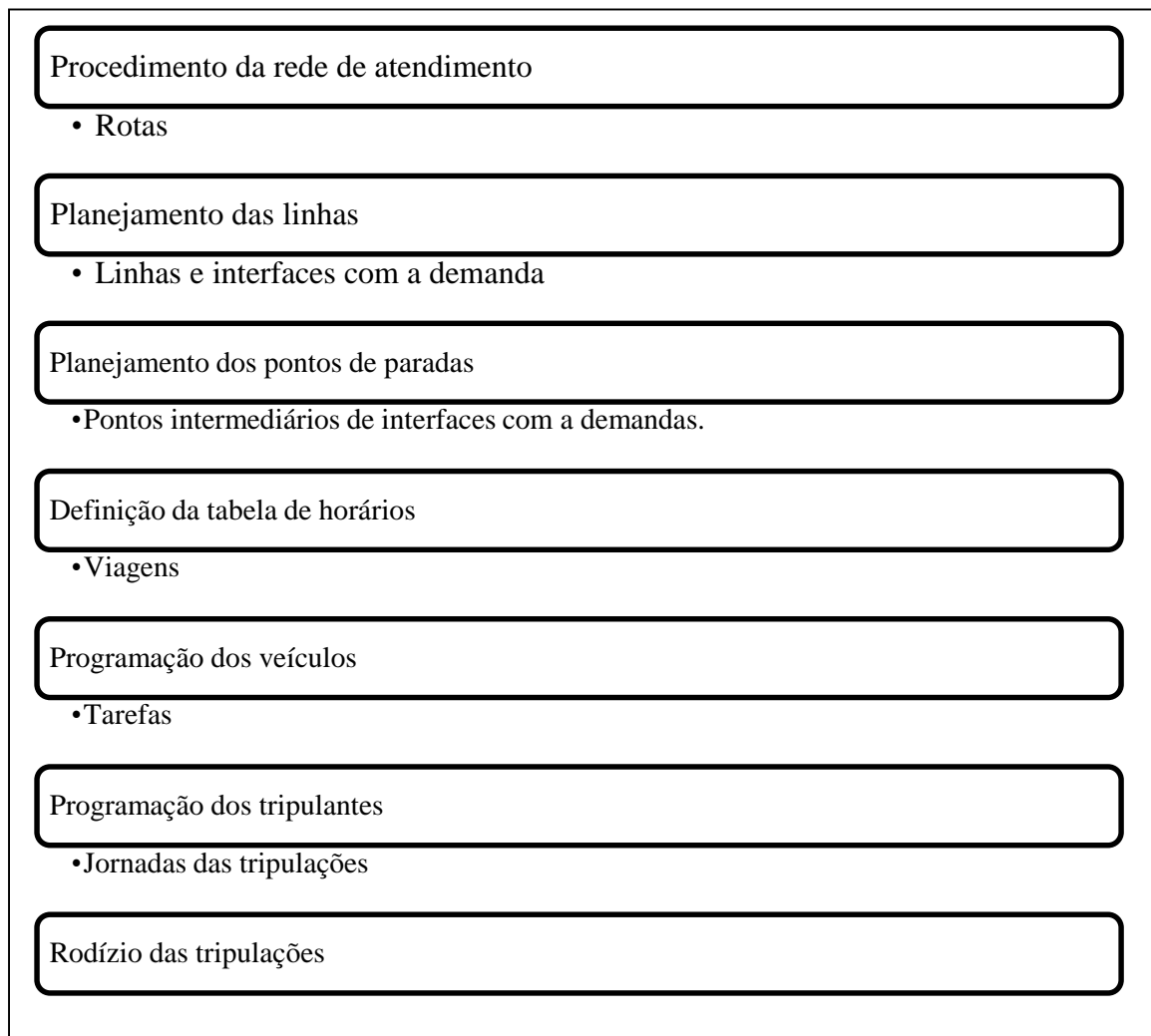
## **2.1 O PROBLEMA DE PROGRAMAÇÃO DE TRIPULAÇÕES**

O planejamento de um sistema de transporte público, em geral, costuma ser dividido em algumas etapas por questões de simplificação.

A etapa inicial, segundo Silva e Silva (2015), consiste na elaboração das rotas e dos quadros de horários, de maneira que todas as regiões da cidade sejam atendidas e que o tempo de espera do usuário nos pontos e terminais atenda a um dado nível de qualidade estipulado no processo licitatório. Segundo esses autores, a segunda etapa considera a alocação dos veículos que executarão as viagens. Nesta etapa, é determinada a quantidade necessária de veículos para atender todas as viagens e rotas, bem como a forma como serão alocadas as viagens aos veículos. A terceira etapa deve determinar a alocação das tripulações, composta por motoristas e cobradores, quando necessário, tomando como base as viagens alocadas a

cada veículo da frota em operação. Nesta etapa, se determina a quantidade necessária de tripulantes e de que forma serão alocadas as viagens aos tripulantes, definindo, portanto, as jornadas diárias de trabalho. A etapa final do planejamento consiste em determinar as escalas mensais da tripulação, ou seja, como é feito o rodízio das tripulações de forma a minimizar os custos e atender regras trabalhistas e operacionais. A Figura 2 evidencia o processo para o planejamento de transportes públicos.

Figura 2: Processo de planejamento de transportes públicos.



Fonte: Adaptado de Reis (2008).

Neste trabalho será considerada a terceira etapa mencionada anteriormente, ou seja o problema de programação de tripulação - PPT, conhecido na literatura como crew scheduling problem. O PPT consiste em determinar o número mínimo de tripulantes e especificar suas viagens de tal forma a cobrir todas as viagens da frota em operação com o menor custo possível (SILVA; REIS, 2014). É nessa etapa que são definidas as jornadas de cada



tripulante, portanto, um tripulante está associado a uma jornada diária de trabalho e vice-versa. Com o intuito de atender as demandas que ocorrem especificamente nos horários de pico as jornadas podem ser de pegada simples, quando o tempo de espera entre as viagens é inferior a duas horas, ou de dupla pegada, quando o intervalo ultrapassa duas horas. Os tripulantes podem realizar uma troca de veículos durante uma jornada de trabalho, o que pode ser uma forma de reduzir o tempo ocioso das tripulações. Entretanto, a empresa restringe o número de trocas realizadas pelas tripulações, normalmente, para facilitar o controle sobre o desgaste dos veículos em operação.

Uma jornada deve ser definida de tal forma que o tempo de descanso entre o seu final e seu início no dia seguinte seja maior ou igual ao tempo mínimo de descanso estipulado pela legislação. Assim, uma mesma tripulação pode realizá-la em dias consecutivos sem infringir a regra do descanso em casa. A jornada também pode ser classificada em dois tipos, sendo nomeados como inteira jornada e meia jornada. A jornada inteira caracteriza-se por um período total de trabalho de 07h20min com um intervalo obrigatório para almoço de uma a duas horas e pode-se acrescentar duas horas a mais, classificadas como hora extra. A meia jornada adota uma jornada de quatro horas totais, não há intervalo para almoço e também para a extensão de hora extra.

Em cada viagem há um custo associado aos salários fixos, horas extras ou ociosas durante a jornada de cada motorista e os custos artificiais como a troca de veículos ou uma dupla jornada. A melhor opção de escala deverá busca balancear o total de horas extras e ociosas dentro do horizonte de planejamento, assim como minimizar o número de total de motoristas.

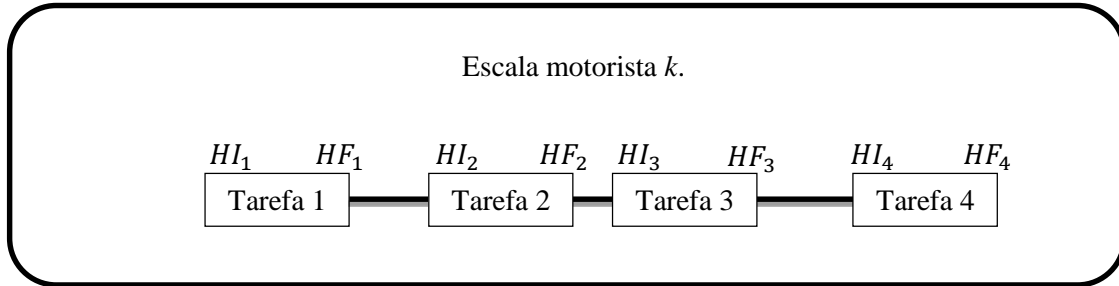
## **2.2 MODELAGEM MATEMÁTICA**

Grande parte dos custos das empresas de transporte público provém da folha de pagamento das tripulações (ROCHA, 2012). A seguir será apresentada uma modelagem matemática para diminuir o gasto com essas tripulações.

Para apresentar a modelagem matemática, faz-se necessário o entendimento dos dados de entrada e de saída do problema.

Uma jornada pode ser representada como na Figura 3, sendo  $HI_j$  a hora de saída da tarefa e  $HF_j$  a hora de chegada da tarefa.

Figura 3: Representação de uma escala



Fonte: Elaboração própria (2016).

Considerando-se as tarefas (viagens)  $i$  e  $j$ , define-se as variáveis de decisão binárias  $y_i^k$ ,  $w_{ij}^k$  e  $x_{ij}^k$  como a seguir:

$$y_i^k = \begin{cases} 1, & \text{se a tripulação } k \text{ realiza a viagem } i \\ 0, & \text{caso contrário} \end{cases}$$

$$w_{ij}^k = \begin{cases} 1, & \text{se a tripulação } k \text{ realiza a viagem } j \text{ imediatamente depois da viagem } i \\ 0, & \text{caso contrário} \end{cases}$$

$$x_{ij}^k = \begin{cases} 1, & \text{se a tripulação } k \text{ realiza uma ou mais viagens entre as viagens } i \text{ e } j \\ 0, & \text{caso contrário} \end{cases}$$

Definindo-se o parâmetro  $HO_k$  como o total de horas ociosas da tripulação  $k$ , o modelo matemático do PPT para  $m$  tarefas e  $n$  motoristas pode ser então representado como:

$$\text{Min } Z = \sum_{k=1}^m HO_k \quad (2.1)$$

Sujeito às restrições:

$$\sum_{k=1}^m y_i^k = 1 \quad \forall i = 1, \dots, n \quad (2.2)$$

$$y_i^k + y_j^k \leq 1 \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.3)$$

com  $j > i$  e  $hc_i > hs_j$

$$\sum_{l=i+1}^{j-1} y_l^k \leq x_{ij}^k * (j - i - 1) \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.4)$$

com  $j > i + 1$

$$\sum_{l=i+1}^{j-1} y_l^k \geq x_{ij}^k \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.5)$$

com  $j > i + 1$

$$w_{ij}^k \leq y_i^k \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.6)$$

com  $j > i$

$$w_{ij}^k \geq y_i^k - x_{ij}^k - 1 \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.7)$$

com  $j > i$

$$w_{ij}^k \leq 1 - x_{ij}^k - 1 \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.8)$$

com  $j > i$

$$\sum_{i=1}^{n-1} \sum_{\substack{j=i+1 \\ 1 \leq hs_i - hc_i \leq 2}}^n w_{ij}^k = 1 \quad \forall k = 1, \dots, m \quad (2.9)$$

$$\sum_{i=1}^{n-1} \sum_{\substack{j=l+1 \\ hs_i - hc_i > 2}}^n w_{ij}^k = 0 \quad \forall k = 1, \dots, m \quad (2.10)$$

$$(hs_i - hc_i) * (y_i^k + y_j^k - 1) - In_k \leq 9,20 - Tm \quad (2.11)$$

$$HE_k \geq \left( (hc_i - hs_i) + (y_i^k + y_j^k - 1) - Intervalo_k \right) - (7,20) \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.12)$$

com  $j > i$

$$x_{ij}^k \in \{0,1\} \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.13)$$

$$y_i^k \in \{0,1\} \quad \forall k = 1, \dots, m, \forall i = 1, \dots, n \quad (2.14)$$

$$w_{ij}^k \in \{0,1\} \quad \forall k = 1, \dots, m, \forall i, j = 1, \dots, n \quad (2.15)$$

$$HE_k \geq 0 \quad \forall k = 1, \dots, m \quad (2.16)$$

$$HO_k \geq 0 \quad \forall k = 1, \dots, m \quad (2.17)$$

A equação (2.1) representa a função objetivo que minimiza a soma de horas extras e tempo ocioso das tripulações. Considera-se hora extra qualquer excedente a 07h20min. Por exemplo, se a tripulação  $k$  realizar uma jornada de 07h40min, terá trabalhado 20 minutos em horas extras.

As restrições (2.2) garantem que todas as viagens sejam realizadas por apenas uma

tripulação, ou seja, para cada viagem  $i$  deve existir, exatamente uma tripulação  $k$  que a realize.

Em (2.3) garante-se que cada tripulação  $k$  só realizará a próxima viagem  $j$  depois de ter realizado a viagem anterior  $i$ . Ressalta-se que a tripulação  $k$  somente poderá realizar a tarefa  $j$  após a tarefa  $i$  se o horário de chegada da viagem  $i$  é menor que o horário de saída da viagem  $j$ , ou seja, se  $hc_i < hs_j$ .

As restrições (2.4), (2.5), (2.6), (2.7) e (2.8) asseguram que a tripulação  $k$  só poderá realizar a tarefa  $l$  entre as tarefas  $i$  e  $j$  somente se também tiver executado a tarefa  $i$  e  $j$ .

Em (2.9) as equações garantem que cada tripulação  $k$  tenha somente um intervalo de repouso e/ou alimentação, de no mínimo uma hora de no máximo duas horas na sua jornada de diária de trabalho.

As equações em (2.10) garantem que cada tripulação não tenha intervalos superiores a duas horas na jornada.

As restrições (2.11) garantem que nenhuma jornada de cada tripulação  $k$  ultrapasse a jornada máxima permitida que seja de 09h20min e em (2.12) é limitado a quantidade máxima de horas extras permitidas.

As equações de (2.13) a (2.17) restringem o tipo das variáveis de decisão.

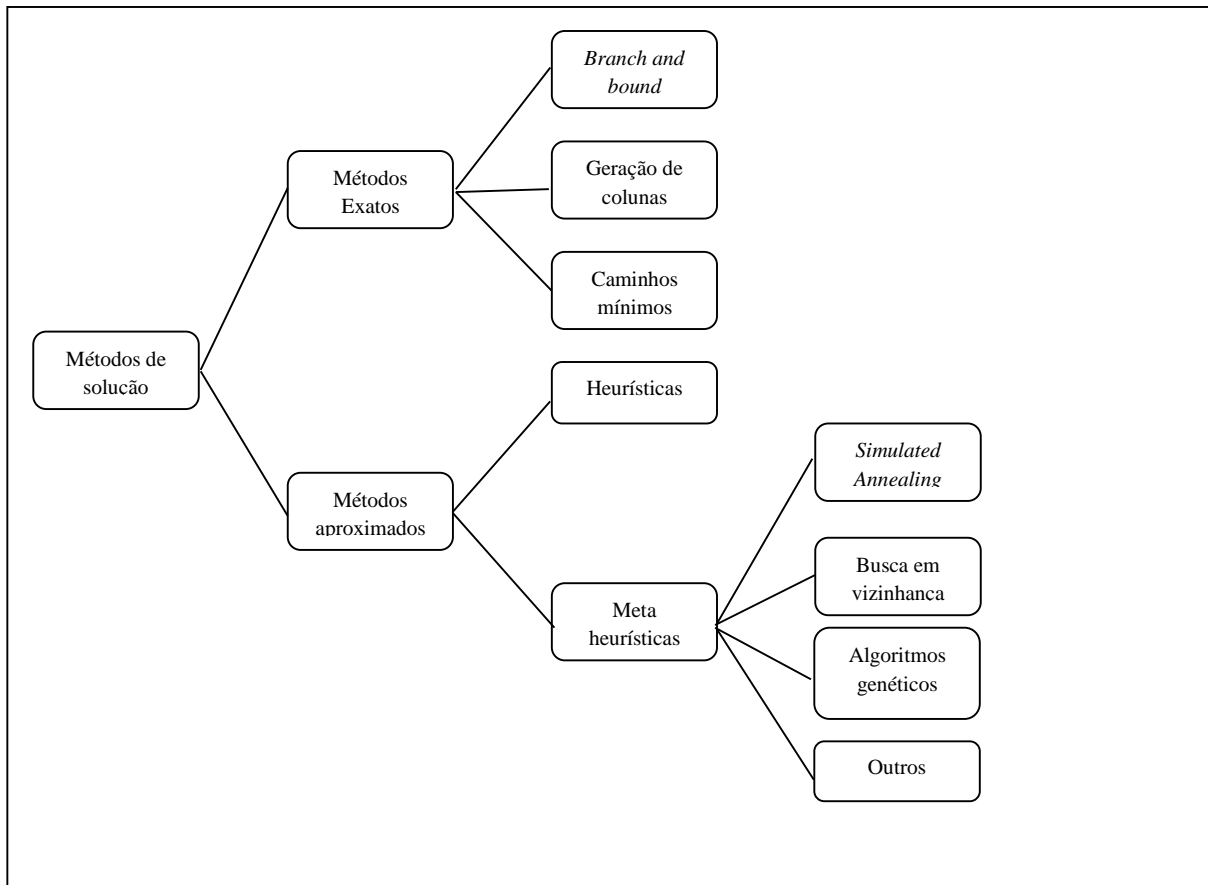
Esta modelagem matemática apresentada foi adaptada para este estudo de caso de Rocha (2012).

## 2.3 MÉTODOS DE SOLUÇÃO

Desejando obter uma solução para o problema de PPT diferentes algoritmos podem ser aplicados.

Em geral, estes algoritmos de otimização são divididos entre métodos exatos e métodos aproximados, sendo o último dividido entre as heurísticas e meta heurísticas, como especificado na Figura 4.

Figura 4: Métodos de solução.



Fonte: Adaptado de Hillier e Lieberman (2013).

### 2.3.1 MÉTODOS EXATOS

Métodos exatos ou métodos de otimização clássica, usam técnicas de decomposição matemática para encontrar a solução ótima do problema de otimização e são muito eficientes em problemas de pequeno e médio porte, porém para sistemas maiores apresentam um elevado esforço computacional e problemas de convergência (TAGLIALENHA, 2008).

O modelo matemático (2.1)-(2.17) proposto neste trabalho permite resolver o PPT de forma exata, porém para um número pequeno de tarefas, uma vez que a quantidade de restrições cresce muito rapidamente a medida que se aumenta o número de tarefas.

Segundo Stefanello (2011) os métodos exatos são mais eficientes para problemas de pequena dimensão ou que não são classificados como NP-completos. Para problemas com grande dimensão, o tempo de implementação tende a aumentar exponencialmente a cada

iteração, então evita-se aplicar modelos exatos. Mesmo assim, existem abordagens exatas para a resolução do problema, entre elas a programação dinâmica, programação linear e inteira, programação por restrições, geração de colunas, branch and bound e relaxação lagrangeana.

Branch and bound por Hillier e Lieberman (2013) é caracterizado pela ideia de dividir para conquistar. Como o problema original é de difícil resolução, ele é dividido em subproblemas cada vez menores até que sejam resolvidos. A avaliação é realizada parcialmente, limitando-se em quão boa pode ser a melhor solução no subconjunto e, a seguir, descartando o subconjunto caso seu limite indique que, possivelmente, ele não possa conter uma solução ótima para o problema original.

Lachtermacher (2009) explica o método como a construção de uma árvore em que os nós representam os problemas candidatos e os caminhos as restrições que devem ser consideradas. Por esta árvore todas as soluções inteiras do conjunto viável serão enumeradas garantido que todas as soluções ótimas serão encontradas. Embora o algoritmo seja conceitualmente simples, apresenta uma complexidade na implementação computacional, tendo excelente desempenho para sistemas pequenos, mas que em sistemas reais apresenta limitações relacionadas ao esforço computacional.

Freling et al (2004) utiliza esta abordagem para encontrar uma solução ótima para o problema de alocação de frota e para o problema de rodízio de tripulação. Utilizando o método de geração de colunas é escolhido um conjunto de jornadas (colunas) para então relaxar o problema na condição da integralidade da solução. Na sequência, são geradas novas colunas possíveis de serem inseridas no conjunto de solução, se apresentarem uma redução de custo. Esta etapa é repetida até que não haja nenhuma jornada que reduza o custo. Caso nesta etapa não se encontre nenhuma coluna a ser trocada o método branch and bound é realizado. Como início, um nó (solução) é gerado para a realização do branch, então, calcula-se os novos limites inferiores das duas soluções, uma gerada na resolução do problema com conjunto de jornadas e a outra após o processo de geração de colunas. Após verifica-se a redução do custo. Depois é verificado se todas as restrições estão sendo respeitadas e se recomeça o processo até que não haja mais uma diminuição no custo. A cada nó é importante considerar uma função a ser utilizada para encontrar o limite inferior do problema no conjunto de nós da árvore de busca. Assim, outro método que pode ser utilizado em conjunto é a geração de colunas em cada limite inferior calculado. A técnica utilizada junto ao branch and bound tem o nome de branch and price.

Em Caprata et al. (2003) o método de branch and price também é abordado para o problema de programação e escalonamento de pessoal. Na primeira etapa foi considerado como um problema de cobertura aplicando métodos de PLI e algoritmos enumerativos, porém a solução encontrada foi pelo branch and price. Em uma segunda etapa foi considerado como um problema de viabilidade, sendo resolvido heurísticamente através de uma sequência de problemas de transporte. Os resultados encontrados foram satisfatórios determinando a melhor solução de casos envolvendo centenas de empregados em um período de 6 meses.

Desrocher e Soumis (1989) abordam o problema de programação de tripulação pela geração de colunas. Para um melhor resultado o problema é dividido em duas etapas, a primeira como um problema de cobertura de conjunto e a segunda como um problema de caminho mínimo com limitação de recurso. Porém, a segunda etapa utiliza a solução encontrada na primeira etapa para realizar uma melhora. Na primeira etapa, o método de geração de colunas é representado por linhas e colunas, sendo as colunas para representar uma escala viável e as linhas, as tarefas a serem realizadas. O modelo seleciona as melhores escalas de modo que todas as tarefas de um dia de trabalho sejam atendidas. Todavia, as novas escalas são geradas pelo algoritmo do caminho mínimo com restrição de recurso. Uma nova coluna é gerada para cada caminho viável, ou seja, uma jornada de trabalho. O modelo proposto obtém resultados satisfatórios para problemas com até 235 tarefas.

Em Santos et al. (2007) também é utilizada a abordagem de geração de colunas, porém combinada a meta heurística de algoritmos genéticos. Na etapa de geração de colunas divide-se o problema em duas partes: problema mestre e subproblema. O primeiro seleciona as jornadas que serão realizadas pela tripulação em um conjunto de jornadas viáveis, de modo a conseguir cobrir todas as viagens necessárias. O subproblema se torna responsável por gerar novas jornadas (colunas). Estas jornadas são adicionadas ao problema mestre se possibilitarem a melhoria na solução. A resolução do subproblema é feita pelo algoritmo genético. Os resultados obtidos com a utilização deste algoritmo se destacam em comparação às outras meta heurísticas pela possibilidade de geração de um conjunto maior de jornadas viáveis para a inclusão no problema mestre e consequentemente possibilita reduzir o tempo de processamento quando lida com grandes instâncias.

Mayrink e Silva (2013) resolvem o PPT em duas etapas. A primeira para determinar a sequência de problemas na designação das jornadas diárias, tentando minimizar a quantidade de horas ociosas e extras compridas pela tripulação. A segunda etapa se torna responsável pela determinação de folgas aos funcionários, visto que para esta etapa é necessário a solução da primeira etapa junto aos sucessivos problemas de designação. O objetivo da mesma é a

redução do número de tripulantes que trabalham para cobrir as tarefas de quem esteja de folga devido a algum direito. O modelo foi aplicado em duas empresas do transporte público da cidade de Belo Horizonte obtendo-se bons resultados para ambas as empresas. Comparando com a solução inicial utilizada na empresa a primeira teve um ganho no corte de motoristas, uma diminuição de 44,54% no total de horas extras e 75,4% no total de tempo ocioso. Para a segunda empresa o ganho foi menor, uma redução de 8,46% de motoristas, 13% no total de horas extras e todas as horas ociosas foram eliminadas.

No trabalho de Mesquita et al. (2011) o PPT é resolvido por um modelo de programação por metas binárias e não lineares, integrando as soluções viáveis entre os veículos e tripulações. A aproximação entre os problemas de programação de veículos (PPV) e ao programa de programação de tripulações (PPT) é dividido três partes, o pré-processamento, resolução da relaxação linear e a obtenção das soluções factíveis. A partir das soluções factíveis, consideradas como solução inicial, a programação de tripulações é finalizada em um contexto não cíclico por um modelo binário biobjetivo considerando os interesses da empresa prestadora de serviço e de seus motoristas. São variados alguns parâmetros do modelo controlados pelo usuário a fim de produzir diferentes soluções.

Bianco et al. (1992) formula o problema através da programação linear inteira, e utiliza uma heurística com um limite inferior proveniente da relaxação da integralidade do modelo, a fim de reduzir as dimensões do problema.

No trabalho de Ernst et al (2004) há uma revisão bibliográfica com diversas referências sobre o problema de programação e rodízio de funcionários, com diversas maneiras de resolução do PPT. Porém, afirma que uma das abordagens mais utilizadas ao resolver PPT é aquela que o decompõem em três etapas: geração das jornadas, otimização das jornadas e definição do rodízio de tripulação. Fazendo assim a complexidade do problema ser reduzida, permitindo uma resolução de forma mais satisfatória.

Fores et al. (1999) apresenta um modelo baseado em PLI como melhoria do sistema TRACS II desenvolvido na Universidade de Leeds, Inglaterra. O TRACS II utiliza duas funções objetivos para calcular as penalidades impostas na solução, assim o modelo proposto utiliza uma função objetivo composta e o método de geração de colunas a fim de obter a solução final. O novo modelo se mostrou melhor que o TRACS II e teve uma redução média de 41% no tempo de processamento para o mesmo conjunto de dados.

Mesmo sendo classificado como um problema NP-difícil, verificou-se diversas formas para gerar programações para tripulação, obtendo soluções otimizadas e com melhorias significantes ao custo salarial às empresas.



### 2.3.1.1 CAMINHOS MÍNIMOS E O ALGORITMO DE DIJKSTRA

Hillier e Lieberman (2013) apresentam o problema de caminhos mínimos como uma rede conectada e não direcionada com dois nós especiais nomeados de origem e destino. A cada uma das ligações, os arcos, está associada uma distância não negativa. O objetivo é encontrar o caminho mínimo, que liga o nó origem ao no destino com a distância total mínima. A essência deste procedimento é que ele se espalha em todas as direções a partir da origem, identificando sucessivamente o caminho mínimo para cada um dos nós da rede na ordem ascendente de suas distâncias (mais curtas) a partir da origem e, assim solucionando o problema quando o nó de destino é atingido.

As conexões ou os arcos podem também representar atividades de algum outro tipo. Portanto escolher o caminho mínimo pela rede corresponde a selecionar a melhor sequência de atividades. Os números que fornecem os “comprimentos” das ligações podem ser, por exemplo, os custos das atividades em cujo caso o objetivo seria determinar qual sequência de atividades minimiza o custo total.

Goldbarg e Luna (2005) afirmam que o caminho mínimo entre dois vértices  $u$  e  $v$  do grafo  $G=(N,A)$ , é uma sequência de arestas que, passando por vértices intermediários liga  $u$  a  $v$  de forma a acumular o menor comprimento ou distância. Para que possa haver um caminho mais curto entre os vértices  $u$  e  $v$  é indispensável que exista primeiramente uma conexão entre  $u$  e  $v$ , ou seja, se existe um caminho de  $u$  para  $v$  direto ou através de outros vértices. Assim o problema de caminho mínimo está intimamente relacionado à solução de vários problemas combinatórios como os de roteamento, programação e sequenciamento de tarefas.

Ball e Benoit-Thompson (1998) abordam o PPT com um conjunto de caminhos que representam as jornadas a serem cumpridas em um grafo. Neste grafo, os nós são as oportunidades de troca e os arcos são de duas classificações: pedaços de jornadas e conexões factíveis entre tarefas. Para incluir as restrições adicionais ao modelo de fluxo de rede, adotasse uma abordagem de relaxação lagrangeana.

Existem várias abordagens à solução do problema de caminhos mínimos, algoritmos com especialização do simplex e algoritmos de fluxo. Porém os mais eficientes hoje estão disponíveis através da abordagem em grafos e duas dominantes são os algoritmos de ajustes sucessivos e por indução e ajuste (FREDERICKSON, 1987 e CHEN, 1990 apud

GOLDBARG e LUNA, 2005).

O algoritmo de Ford-Moore-Bellman trabalha com os arcos do grafo, ou seja, vai procurá-los, um após o outro, em uma ordem determinada para ver se algum deles melhora o caminho entre o vértice de origem ao de destino. Seu critério de parada é quando em uma rodada de atualizações com todos os arcos não mostrar nenhuma melhoria. Fornecendo sua resposta no máximo em  $n$  iterações, sendo  $n$  o número de vértices do grafo, assim sua complexidade computacional é classificada em  $O(mn)$  (BOAVENTURA NETTO, 2009).

Em 1959 Edsger Wybe Dijkstra sugeriu um algoritmo de rotulação para caminhos em grafos com arcos positivos, utilizando indução e ajuste, de forma eficiente e de fácil implementação computacional. O algoritmo de Dijkstra calcula as menores distâncias de um vértice inicial a todos os outros (Figura 5) considerando as seguintes premissas:

- $C$ : matriz de custos representando as distâncias entre vértices ligados diretamente;
- $s$ : nó de origem;
- $t$ : nó de destino;
- $dist$ : vetor que contém os custos;
- $prev$ : vetor que contém os sucessores.

Figura 5: Algoritmo de Dijkstra.

```

Algoritmo de Dijkstra  $D = (G, s, t)$ 
Início
Crie um conjunto vazio  $Q$ 
   $dist(s) = 0$ 
Para cada vértice  $v \in G$ 
   $dist[v] = infinito$ 
   $prev[v] = \emptyset$ 
  Insira  $v$  em  $Q$ 
Enquanto  $Q$  não for vazio
   $u =$  vértice em  $Q$  com menor valor de  $dist[u]$ 
  Remova  $u$  de  $Q$ 
  Se  $u$  igual a  $t$ 
    Encerre a execução e retorne  $dist$  e  $prev$ 
  Para cada vizinho  $v$  de  $u$ , tal que  $v \in G$ 
    Se  $dist[u] + C[u, v] < dist[v]$ 
       $dist[v] = dist[u] + C[u, v]$ 
       $prev[v] = u$ 

```

Fonte: Adaptado de Boaventura Netto (2009).

Como demonstrado na Figura 5, o rótulo (*prev*) colocado em cada vértice guarda a sua origem de posição e em *dist* é guardado o valor do caminho acumulado até o vértice. A cada iteração o nó examinado que acumula a menor distância é fechado, ou seja, removido da lista *Q*, para então examinar os vizinhos dele a fim de atualizar os valores em *dist* e *prev* quando possível (GOLDBARG e LUNA, 2005).

O algoritmo apresenta uma complexidade computacional de  $O(n^2)$  (quando na implementação não é usado uma fila de prioridade) e não é capaz de encontrar caminhos mínimos em presença de arcos negativos, uma vez que a cada iteração o vértice examinado com menor distância acumulada é fechado.

### 2.3.2 MÉTODOS APROXIMADOS

Quando não é possível determinar a solução exata de um dado problema de otimização é comum se recorrer ao uso de métodos alternativos, que em geral podem determinar uma solução aproximada para o problema em um tempo computacional razoável. Esses métodos alternativos são conhecidos na literatura como métodos heurísticos ou meta heurísticas, dependendo de sua complexidade de implementação.

Como mencionado no capítulo introdutório deste trabalho, uma heurística consiste em um método de resolver problemas através de técnicas práticas aprendidas em experiências passadas; formada da palavra grega “heuriskein”, que significa descobrir, inventar, ter uma ideia, é utilizado para descrever todas essas técnicas que, ao invés de usar uma aproximação da otimização clássica, gera, passo a passo, uma solução avaliando e selecionando opções, com ou sem a ajuda do planejador. Para isto, nos modelos heurísticos realizam-se buscas locais com a orientação lógica ou empírica de índices de sensibilidade (regras heurísticas) (TAGLIALENHA, 2008).

Estas regras são utilizadas para gerar e classificar as opções de construção de uma solução de boa qualidade durante a busca. Em geral, estes métodos apresentam a vantagem de fornecer soluções de boa qualidade com esforços computacionais moderados, mas também não garantem que se encontre a solução ótima de sistemas reais e não fornecem informações sobre a qualidade da solução obtida. Entretanto, estes métodos ainda são os mais utilizados para problemas de alta complexidade.

De uma maneira geral os métodos heurísticos se baseiam em ideias relativamente

simples de senso comum ao procurar uma boa solução. Eles tendem a ser específicos para cada caso, ou seja, geralmente são desenvolvidos para atender a um tipo de problema em vez de uma variedade de aplicações (HILLIER E LIEBERMAN, 2013).

Tagliarenha (2008) destaca que os métodos heurísticos geram a cada iteração uma nova solução que avalia e seleciona as opções possíveis, tendo ou não o auxílio de um planejador. Isto ocorre a partir de buscas locais com uma indicação lógica ou empírica de índices de sensibilidade (regras heurísticas). De uma maneira geral as soluções obtidas são de boa qualidade e com esforço computacional menor, porém como já dito é possível garantir que seja obtida uma solução ótima.

As heurísticas de melhoramento também se destacam para a programação. Lin (1965) se baseia no procedimento de K-opt, dividindo o processamento em K segmentos. O procedimento é realizado a partir da nova conexão entre os segmentos previamente divididos, sendo aceito se melhorar o custo e como regra de parada assim que não for possível melhorar as rotas. A complexidade do algoritmo é de  $O(n^k)$ , sendo  $n$  o número de entradas de dados e  $k$  o número de trocas entre as arestas.

Alvarenga (2005) comenta que a utilização de heurísticas clássicas foram perdendo espaço para as meta heurísticas, porém ainda são usadas de forma complementar, como por exemplo, para determinar a solução inicial em alguns algoritmos. Outra importância está no fato de que permanecem como operadores de outros métodos baseados em busca local, são responsáveis por encontrar grupos de soluções a serem avaliadas, também conhecido como vizinhança. Tem como objetivo de encontrar soluções de melhor qualidade e em um custo computacional reduzido por uma combinação de métodos heurísticos.

“Uma meta heurística é um tipo de resolução geral que orchestra a interação entre procedimentos de melhoria local e estratégias de nível mais alto para criar um processo que seja capaz de escapar dos ótimos locais e realizar uma busca consistente de uma região de soluções viáveis.  
“HILLIER; LIEBERMAN, 2013.

Entre as meta heurísticas existentes destacam-se simulated annealing, busca em vizinhança variável, algoritmo genético, busca tabu e alguns trabalhos são destacados a seguir.

Simulated annealing de acordo com Kirkpatrick et al. (1983) é classificado como o recozimento (annealing) de alguns materiais. Consiste em submetê-los inicialmente em altas temperaturas e reduzi-las gradualmente até atingirem, com o aumento e ou redução do estado

de energia, o equilíbrio térmico, tornando-os assim consistentes e rígidos, ou seja, a produção de cristais perfeitos. Kirkpatrick et al.(1983) teve a ideia de aplicar este método para resolver problemas de otimização combinatória. A analogia do processo para o recozimento de metais com um problema combinatório pode ser descrita como:

- Os estados possíveis de um metal correspondem a soluções do espaço de busca;
- A energia em cada estado corresponde ao valor da função objetivo;
- A energia mínima ou máxima, depende da condição do problema: minimização ou maximização, corresponde ao valor de uma solução ótima local, possivelmente global.

Tagliailenha (2008) também descreve a utilização do procedimento em quatro passos:

- Identificar a função energia do sistema com a função objetivo que se quer otimizar;
- Associar os átomos do sistema às variáveis do problema;
- Para cada temperatura de uma sequência de temperaturas decrescente realiza-se a simulação descrita, ou seja, a cada iteração um novo estado é gerado a partir do estado atual por uma modificação aleatória;
- No final do processo, espera-se que o sistema estacione em um estado de energia globalmente mínima, por analogia com a física do problema.

Emden-Weinert; Proksch (1999) apresenta-se uma aplicação do simulated annealing no problema de alocação de tripulação de linha aérea baseado em uma formulação de corte de corrida. Os resultados computacionais são relatados para alguns problemas de teste de curto e médio alcance do mundo real com até 4600 voos por mês. Os autores concluem que o tempo de execução pode ser economizado e a qualidade da solução pode ser melhorada usando uma solução inicial específica para o problema, aliviando as restrições e combinando o recozimento simulado com uma heurística de melhoria local específica do problema e múltiplas execuções independentes.

As vantagens do modelo segundo Bueno (2009) são que o procedimento do simulated annealing tem capacidade de resolver problemas de diversos níveis de complexidade e em várias áreas específicas, e por trabalhar com poucos parâmetros envolvendo operações simples de matemática e computacionais apresenta uma relativa previsibilidade e simplicidade.

A meta heurística busca em vizinhança, seja a variable neighborhood search (VNS) ou

variable neighborhood descent (VND), foram propostas por Nedad Mladenovic e Pierre Hansen em 1997. De uma maneira geral ela tenta explorar o espaço de soluções através de trocas sistemáticas de estruturas de vizinhança, sendo assim a meta heurística de busca em vizinhança variável seria uma extensão do algoritmo de busca local que utiliza a estratégia de mudança do tamanho da vizinhança para sair de soluções ótimas locais. Segundo Tagliarenza (2008), um algoritmo de busca local primeiramente deve gerar uma solução inicial  $s$ , depois aplica uma busca local afim de encontrar a melhor solução vizinha  $s'$  e caso  $s'$  seja melhor que  $s$  então a troca é aceita e aplica-se novamente a busca para melhorar a solução a vizinhança da nova solução.

O VND é baseado no fato de que um mínimo local com relação a estrutura de vizinhança não é necessariamente um mínimo local com relação a uma outra vizinhança. Assim, a solução final dada por este algoritmo é um mínimo local em relação a todas as estruturas de vizinhanças e, portanto, a probabilidade de se alcançar o mínimo global é maior de quando se usa somente uma estrutura (MLADENOVIC; HANSEN, 1999, GENDREAU; POTVIN, 2010).

O VNS, não seguindo o padrão de muitas meta heurísticas locais, não permite uma degradação da função objetivo para realizar um movimento, assim ela explora a partir de uma solução inicial uma sequência crescente de vizinhanças, e realiza o movimento até uma nova solução se esta for melhor que a solução incumbente (HANSEN; MLADENOVIC, 2001).

Reis e Silva (2011) utilizam o VNS combinando com o VND e o método de busca em vizinhança de grande porte (VLNS) para solucionar o problema de uma empresa da cidade de Belo Horizonte, Minas Gerais. O VNS é iniciado com uma solução factível e um conjunto de estrutura de vizinhança. Como procedimento de busca local foram utilizados VND e VLNS, e para as implementações foram consideradas que as tripulações só poderiam realizar no máximo uma troca e duas trocas de veículos durante a jornada. Em ambos os métodos os resultados foram melhores quando comparados aos usados pela empresa originalmente.

Os algoritmos genéticos são ideias transferidas a partir da teoria da evolução biológica das espécies formada por Charles Darwin, em que indivíduos com uma variação genética tem uma probabilidade maior de chances de sobrevivência para as gerações seguintes. Hillier e Lieberman (2013) caracterizam que as soluções viáveis para o problema correspondem aos integrantes de dada espécie e que a adaptação de cada um é medida pela função objetivo. A cada iteração uma população é formada através dos conjuntos de soluções experimentais considerados, tendendo a gerar populações mais evoluídas (soluções melhores) à medida que o algoritmo prossegue. Ocasionalmente ocorrem as mutações entre os filhos (novas soluções

geradas), ajudando o algoritmo explorar uma nova, quem sabe melhor, região de soluções possíveis.

Segundo Hillier e Lieberman (2013) o algoritmo deve seguir os seguintes passos:

1. Gerar uma população inicial após escolher a forma de codificação;
2. Calcular a função objetivo de cada configuração e armazenar a solução incumbente;
3. Realizar a seleção;
4. Implementar a recombinação (*crossover*);
5. Implementar a mutação e obter a nova geração;
6. Estabelecer um critério de parada, caso o critério for satisfeito, pare. Caso contrário, voltar ao passo 2.

O algoritmo tem algumas características peculiares em relação aos outros, são baseados em um conjunto de soluções possíveis, não envolvem a modelagem do problema visto que estes podem ter variáveis binárias ou não, mudando drasticamente a modelagem e é um método probabilístico. Assim a mesma população dificilmente apresenta os mesmos resultados para um mesmo problema, (BUENO, 2009).

Li e Kwan (2003) propõem um algoritmo genético híbrido para resolver o problema de programação da tripulação para problemas de ônibus urbano e linhas de trem com dados de empresas inglesas. A partir de uma heurística gulosa foi construída uma escala, em que as jornadas eram escolhidas sequencialmente de um grande conjunto de potenciais jornadas viáveis geradas previamente. Neste, as jornadas individuais e as escalas devem ser avaliadas como um único processo. Nas jornadas individuais, a viabilidade da solução é avaliada a partir de conhecimento prático do problema. Então o algoritmo genético gera uma distribuição de pesos quase ótimos entre os critérios, em que uma avaliação ponderada de um único valor pode ser computada para cada jornada. A programação construída a partir da distribuição dos pesos gerados é avaliada pela função de aptidão do algoritmo genético, considerando o objetivo de minimizar jornadas e minimizar o custo total. Foi observado a partir dos resultados que o algoritmo proposto consegue resolver problemas de tamanhos reais.

Outra meta heurística utilizada na resolução do PPT é a busca tabu (*Tabu Search*, TS), esta utiliza ideias de senso comum ao permitir que o processo de busca escape de um ótimo local. O método também é conhecido por subida íngreme ou descida mais suave (ANDRADE, 2013).

Fred Glover a propôs na década de 80 com o conceito baseado em inteligência

artificial, de uma maneira que a resolução do problema deve ser considerada inteligente se no processo for incorporado uma memória adaptativa e com exploração sensível.

Hillier e Lieberman (2013) explica o algoritmo TS como um procedimento de melhoria local, exceto que as vezes não é necessário que a cada iteração a nova solução seja melhor que a anterior. O processo se inicia com um procedimento de melhoria local da maneira usual, ou seja, aceitando apenas uma solução melhor a cada iteração afim de encontrar um ótimo local. O diferencial do TS é por permitir a busca mesmo tendo movimentações sem melhorias, porém essa abordagem tende que a busca se concentre nas vizinhanças do ótimo local já encontrado. Para isto não ocorrer, é criada uma lista de tabus em que são registradas as movimentações proibidas.

Em Nurmi, Kyngas e Post (2011) é proposto uma solução com a combinação dos métodos de simulated aneling e Busca Tabu à uma empresa finlandesa de transporte urbano. A solução inicial foi criada alocando as jornadas em dias aleatórios. As próximas jornadas são aceitas se causarem uma diminuição na função objetivo. Aplica-se o simulated annealing a fim de evitar a ficar preso em áreas não promissoras. Para adicionar penalidade são atribuídos pesos dinâmicos e restrições difíceis. O algoritmo se mostra eficiente com entradas randômicas de dados.

Marinho et al. (2004) também utiliza a Busca Tabu para resolver o PPT em três tipos de busca, obtendo resultados relevantes ao serem comparados com outras heurísticas abordadas na literatura e com a solução inicial da empresa. Sendo utilizada duas estruturas de vizinhanças, realocação e troca de tarefa entre jornadas, a lista tabu teve como objetivo evitar que a heurística entrasse em um ciclo durante a busca do melhor vizinho. A lista era utilizada afim de armazenar as últimas soluções avaliadas, para que o método não voltasse a solução anteriormente avaliada. As três implementações da Busca Tabu foram: Busca Tabu com o primeiro vizinho de melhora em uma porcentagem de vizinhança; Busca Tabu com o melhor vizinho em uma vizinhança variável e a Busca Tabu com o primeiro vizinho de melhor em uma porcentagem em uma vizinhança com diversificação, esta que reinicia o procedimento se após um número de iterações não encontrar uma solução viável.



### **3 MÉTODO PROPOSTO**

Neste capítulo detalham-se as etapas do método proposto para a obtenção das novas escalas dos motoristas, a avaliação das escalas fornecidas, os parâmetros utilizados na implementação e finaliza-se com a apresentação dos resultados obtidos.

#### **3.1 ESCALA ATUAL DA EMPRESA**

A empresa em estudo forneceu um banco de dados com as informações das escalas atuais praticadas pela mesma. Neste banco de dados, em arquivos no formato de PDF, são detalhadas todas as tarefas a serem cumpridas junto às escalas dos motoristas como mostrado a seguir.

Considerou-se um banco de dados que apresenta as informações relacionadas aos motoristas denominado ‘Projeto escala’ como pode ser observado na Tabela 2, em que:

- Código da linha: Identificador do caminho percorrido pela tarefa;
- Código do veículo: Número identificador do veículo;
- Hora saída: Horário do início da tarefa;
- Hora chegada: Horário do fim da tarefa;
- Ponto inicial: Local de saída da tarefa;
- Ponto final: Local de chegada da tarefa;
- Tipo: Classificação da viagem;
- Motorista: Número identificador do motorista.

Tabela 2: Representação de uma escala.

Código linha	Código veículo	Hora de saída	Hora de chegada	Ponto inicial	Ponto final	Tipo	Motorista
1401	22	03:55	04:20	Barra Velha	Terminal centro	Ida	3001
1401	22	04:28	05:13	Terminal Centro	Terminal centro	Circular	3001
1401	22	05:15	06:00	Terminal centro	Terminal centro	Circular	3001
1401	22	06:05	06:50	Terminal centro	Terminal centro	Circular	3001
1401	22	06:50	07:35	Terminal centro	Terminal centro	Circular	3001
1401	22	07:45	08:25	Terminal centro	Terminal centro	Circular	3001
1401	22	09:47	10:32	Terminal centro	Terminal centro	Circular	3001
1401	22	11:00	11:40	Terminal centro	Terminal centro	Circular	3001

Fonte: Elaboração própria (2016).

No sequenciamento de tarefas utilizadas atualmente pela empresa os parâmetros como número de motoristas, tempo total ocioso e tempo total de horas extra foram avaliados e apresentados na Tabela 3.

Tabela 3: Avaliação das escalas utilizada pela empresa.

Parâmetros	Motoristas de inteira jornada	Motoristas de meia jornada	Total tempo ocioso total (minutos)	Total hora extra (minutos)
Total	32	0	2381	65

Fonte: Elaboração própria (2016).

Para a obtenção de uma proposta de solução neste trabalho considerou-se as tarefas indicadas no banco de dados, com um total de 343 tarefas.

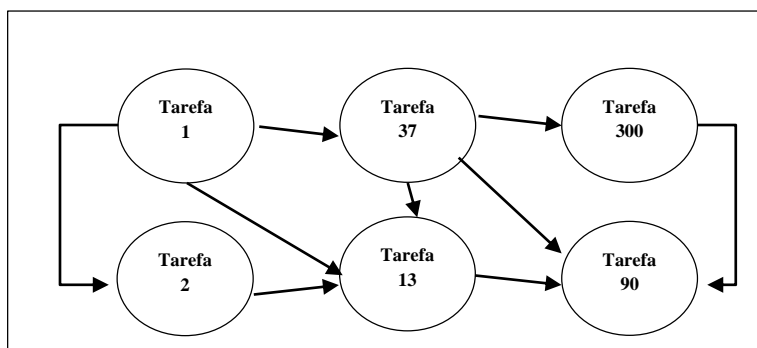
### 3.2 ABORDAGEM PROPOSTA

Na construção do sequenciamento das tarefas será considerada a abordagem de caminhos mínimos. Essa ideia mostra-se atrativa visto que o caminho mínimo fornece o menor custo entre um nó de origem e um nó de destino.

O caminho mínimo será construído baseado em um grafo obtido a partir das tarefas existentes. Cada nó do grafo representa uma tarefa a ser alocada a uma escala, e as arestas do grafo serão criadas quando forem respeitadas as condições para seguir de uma tarefa a outra, ou seja, somente será possível seguir da tarefa atual  $i$  para a próxima tarefa  $j$ , quando o ponto de chegada da tarefa  $i$  for igual ao ponto de saída da tarefa  $j$ , e quando o horário de chegada da tarefa  $i$  for menor que o horário de saída da tarefa  $j$ .

O grafo completo representa todas as tarefas que devem ser atribuídas a escala de algum motorista. Devido ao grafo apresentar 343 nós uma pequena parte está demonstrada na Figura 5.

Figura 5: Representação do grafo de tarefas.



Fonte: Elaboração própria.

Através da Figura 5 notasse que a tarefa 1 está conectada diretamente para as tarefas 2, 37 e 13, ou seja, após realizar a tarefa 1 posso fazer a que tiver o menor tempo ocioso entre essas três tarefas. A aresta entre cada nó é somente criada se respeitar as restrições de ponto de chegada e saída e o horário de chegada da tarefa for menor que o horário de saída das demais tarefas. Assim o ponto de chegada da tarefa 1 é igual ao ponto de saída das três tarefas citadas anteriormente e horário de chegada da tarefa 1 é menor que o horário de saída das demais tarefas.

O caminho entre as tarefas será construído considerando o algoritmo de Dijkstra, que fornece o caminho entre o nó de origem ao nó de destino e seu custo (BOAVENTURA NETTO, 2006).

Obtido o caminho entre o nó de origem (primeira tarefa) aos demais nós, determina-se a primeira sequência de tarefas a serem cumpridas pelo motorista antes do intervalo de almoço. O nó de destino será a última tarefa antes do intervalo e este é determinado pelo seu horário de chegada ser o mais próximo de quatro horas trabalhadas.

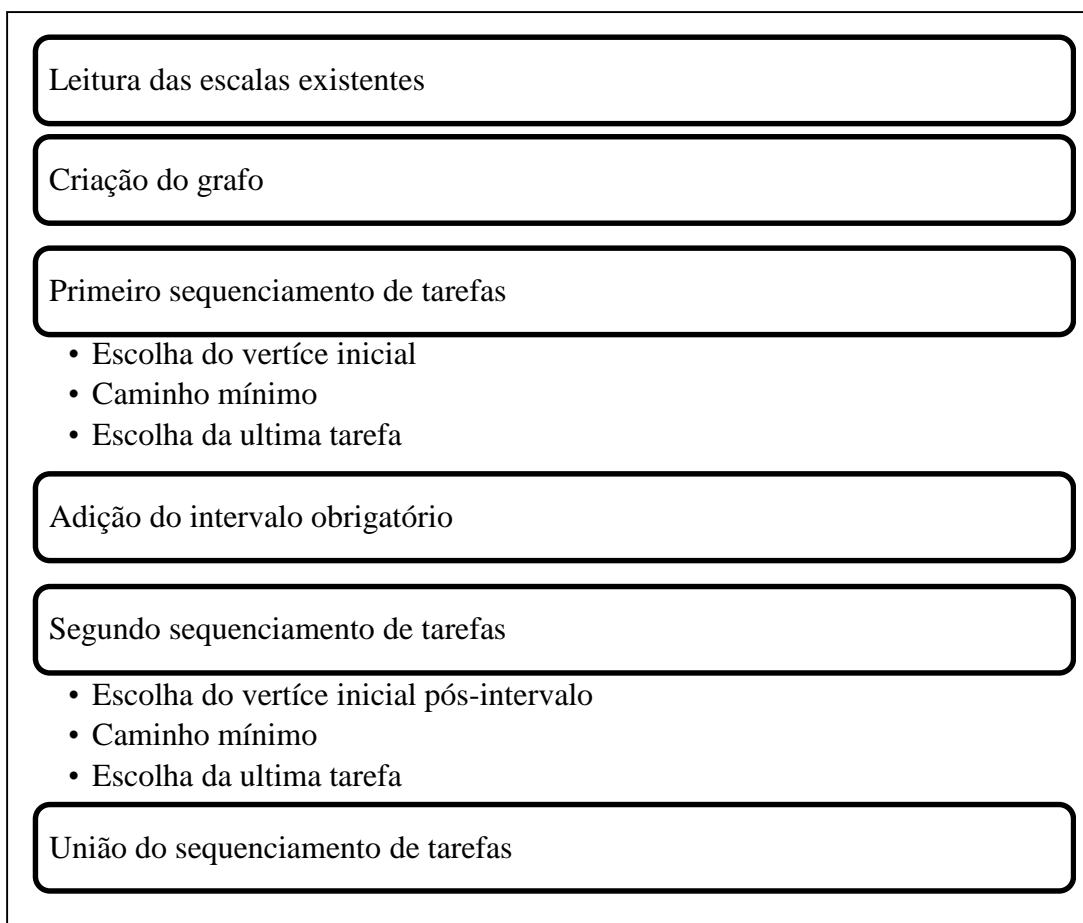
Após a adição do intervalo obrigatório, o procedimento descrito acima é repetido para a construção do segundo sequenciamento de tarefas. O nó de origem é o nó com o horário de saída mais próximo ao valor do final do tempo de intervalo, e o nó de destino é o nó com tempo de chegada mais próximo de completar a jornada, podendo haver a possibilidade de acrescentar hora extra ao motorista.

Se o sequenciamento completo das tarefas respeitar todas as restrições descritas no capítulo anterior, o sequenciamento é aceito e removem-se então do grafo as tarefas já alocadas e repete-se o processo até que todas as tarefas sejam atribuídas a uma escala.

De forma iterativa, constroem-se vários caminhos até que todas as tarefas sejam alocadas aos motoristas, obtendo-se ao final do processo uma solução viável, equivalente a um conjunto de escalas prontas a serem designadas aos motoristas.

O sequenciamento das etapas é destacado na Figura 6.

Figura 6: Sequenciamento da criação de uma escala.



Fonte: Elaboração própria (2016).

Os procedimentos foram implementados no ambiente MATLAB, e serão detalhados na Seção 3.4.

### 3.3 PARÂMETROS

Os parâmetros utilizados na construção do sequenciamento de tarefas são apresentados na Tabela 4. Estes são apresentados para auxiliar no entendimento à implementação do trabalho mostrado no tópico seguinte.

O vértice inicial e o tempo antes do intervalo são parâmetros escolhidos aleatoriamente pelo programa, foram determinados desta maneira para que não houvesse um padrão nas escolhas dos vértices iniciais e que em cada escala o tempo de trabalho antes do

intervalo fossem diferentes para gerar distintos sequenciamentos.

Os demais parâmetros são considerados constantes em toda implementação.

Tabela 4: Parâmetros utilizados na construção do sequenciamento de tarefas.

Parâmetro	Valor	Descrição
Vértice inicial	1 a 343	Tarefa inicial da iteração atual e escolhido aleatoriamente
Hora extra	570	Tempo máximo de hora extra em minutos, a ser usado quando for uma jornada inteira
Tempo antes do intervalo	210 a 270	Tempo considerado para jornada antes do intervalo ou para ser considerado um motorista de meia jornada
Intervalo	70	Tempo adotado para o intervalo de almoço

Fonte: Elaboração própria (2016)..

### 3.4 IMPLEMENTAÇÃO

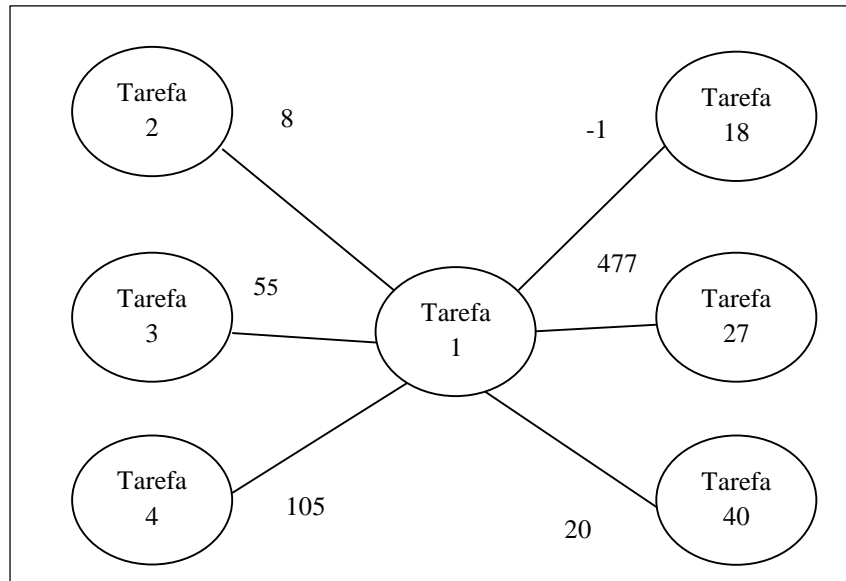
Primeiramente, foi lida a escala atual usada pela empresa para obtenção das informações de todas as tarefas a serem realizadas, de forma que cada tarefa lida corresponde a um novo nó contendo os dados como hora de saída, hora de chegada, ponto de saída e chegada, linha, sentido e identificador.

Considerou-se o custo da conexão (aresta) entre um nó e outro como sendo o tempo ocioso entre cada tarefa, o qual é obtido pela equação (3.1), porém só será atribuído somente se o ponto de chegada da tarefa anterior for igual ao ponto de saída da próxima tarefa.

$$T_{ocioso} = T_{saída(próxima)} - T_{chegada(anterior)} \quad (3.1)$$

A Figura 7 representa uma parte do grafo construído a partir da tarefa 1 às demais tarefas.

Figura 7: Representação do grafo de tarefas.



Fonte: Elaboração própria (2016).

Assim a partir da tarefa 1 pode-se seguir para as tarefas 2, 3, 4, 27 e 40 com o tempo ocioso de 8, 55, 105, 477 e 20 respectivamente. Na tarefa 18 foi atribuído o custo de -1 a aresta, pois como mencionado o algoritmo de Dijkstra não realiza caminho em arestas com custo negativo.

Após todas as tarefas serem lidas pelo programa, foi criada uma matriz  $n \times n$  sendo  $n$  o número de tarefas a serem realizadas. Esta matriz é obtida como a representação da alcançabilidade de cada nó aos demais nós, ou seja, se existe um caminho direto de um nó ao outro. A matriz também é considerada como a representação do grafo na implementação.

O algoritmo é apresentado na Figura 8.

Figura 8: Algoritmo da matriz de alcançabilidade.

Algoritmo de construção da matriz de alcançabilidade

Para cada tarefa existente (atual) em  $G$ :

Se Hora de saída (próxima) > Hora de chegada (atual) e Ponto de saída de (próxima) = ponto de chegada de (atual)

$G(\text{atual}, \text{próxima}) = \text{Hora de Chegada (próxima)} - \text{Hora de saída (atual)}$ .

$G(\text{atual}, \text{próxima}) = -1$ .

Fonte: Elaboração própria (2016).

Tendo respeitado as condições estabelecidas de hora de saída e de chegada e de ponto de saída e de chegada, atribuiu-se o tempo ocioso entre a tarefa atual e a próxima tarefa. Se a condição de teste não é respeitada o valor a ser alocado na matriz é negativo, pois o algoritmo de Dijkstra não realiza caminho entre tarefas com custos negativos.

Também para auxiliar a função de Dijkstra, se a subtração entre a hora de chegada e saída for igual a zero, é atribuído uma contribuição de 0.0001 ao valor, pois o algoritmo não atualiza o caminho se o peso atribuído na aresta for igual a zero ou menor que zero.

A partir da matriz determinada, é possível aplicar o algoritmo de Dijkstra com o propósito de encontrar o caminho mínimo entre uma tarefa e as demais.

Assim constroem-se os caminhos mínimos para vários nós e então dividem-se esses caminhos mínimos em partes para respeitar as condições de trabalhos, sendo elas as restrições matemáticas descritas previamente.

Para a construção de cada solução segue-se os seguintes passos:

- I. Leitura das escalas existentes;
- II. Escolha aleatória do vértice (nó) inicial;
- III. Criação da matriz de alcançabilidade;
- IV. Criação do caminho (*prev*) e custo (*dist*) do vértice inicial pelo algoritmo de Dijkstra.

Na sequência determina-se a última tarefa antes do intervalo obrigatório, este tempo antes do intervalo pode ser adotado entre 3h30min a 4h30min de acordo com as leis trabalhistas e escolhido randomicamente pelo programa.

Pelo vetor de caminhos, obtêm-se o caminho feito entre a última tarefa antes do intervalo até a tarefa inicial, ou seja, obtêm-se a sequência de tarefas de antes do intervalo



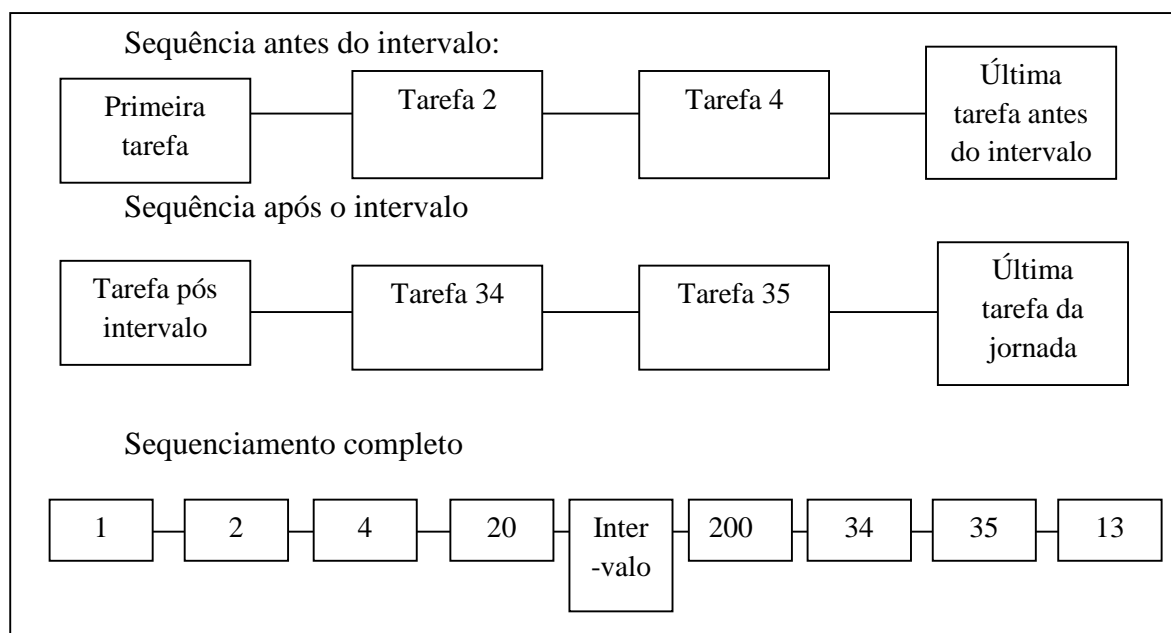
obrigatório que algum motorista irá realizar. A partir do horário de chegada da última tarefa antes do intervalo, soma-se 70 minutos, e procura-se a tarefa que tiver seu tempo de início o mais próximo possível deste horário, visando diminuir ao máximo o tempo ocioso entre o fim do intervalo com a próxima tarefa. Tendo esta tarefa determinada, é feito novamente um vetor de caminhos por Dijkstra a partir desta tarefa.

Após determina-se a última tarefa para o período todo de trabalho, a escolha desta é a que tem o horário de chegada mais próximo de completar a jornada completa de trabalho de 07h20min ou 09h20min por ser permitido trabalhar com horas extras.

Mais uma vez pelo vetor caminho a sequência de tarefas é informada, através do nó de origem como a tarefa pós-intervalo e o nó de destino à última tarefa.

De forma geral, realizam-se duas sequências de tarefas, uma para antes do intervalo e outra para depois do intervalo. Para se obter a escala diária completa basta unir as duas sequências. A Figura 9 demonstra uma sequência produzida.

Figura 9: Construção de uma sequência de tarefas.



Fonte: Elaboração própria (2016).

Antes de começar outra escala, verifica-se se a sequência obtida respeita todas as restrições impostas, ou seja, se é uma escala viável ou não para ser designada a um motorista. Em caso afirmativo, retiram-se do grafo as tarefas que já foram atribuídas para que não haja dois motoristas realizando as mesmas tarefas, como dito na restrição 2 e novamente segue-se os passos I a IV anteriormente.

Caso a sequência não seja viável, a mesma não é considerada uma escala que pode ser executada por algum motorista. Assim os seguintes passos são feitos para se fazer um novo sequenciamento:

- V. Escolha aleatória de um novo vértice inicial;
- VI. Geração de outro sequenciamento de tarefas, tal qual descrito anteriormente.

Este procedimento se repete até que todas as tarefas tenham sido atribuídas a algum motorista, ou seja, quando o tamanho da matriz de alcançabilidade for igual a zero.

Com o intuito de avaliar as escalas geradas, foi criada uma função de avaliação, com os seguintes parâmetros avaliativos:

- Número de motoristas;
- Total do tempo ocioso dos motoristas;
- Total de horas extras dos motoristas.

Para cada conjunto de escalas viáveis aplicou-se a função de avaliação, determinando-se assim o tempo total ocioso de cada motorista, bem como o tempo total de horas extras pagas a cada motorista.

### 3.5 RESULTADOS OBTIDOS

Foram realizadas 1000 execuções da implementação descrita no programa Matlab. As análises foram feitas pela função de avaliação descrita anteriormente.

Com o propósito de escolher a melhor solução, avaliou-se o número de motoristas, número de tarefas que não foram alocadas, tempo total ocioso e tempo de hora extra e determinou-se os seus valores médios e os desvios-padrão, os quais são apresentados na Tabela 5.

Tabela 5: Avaliação dos parâmetros comparativos.

Parâmetro	Número de motoristas	Tarefas não alocadas	Tempo ocioso (minutos)	Horas extra (minutos)
Média	22	51	1916	844,2481
Desvio padrão	2,5044	13,4449	161,8310	331,2481

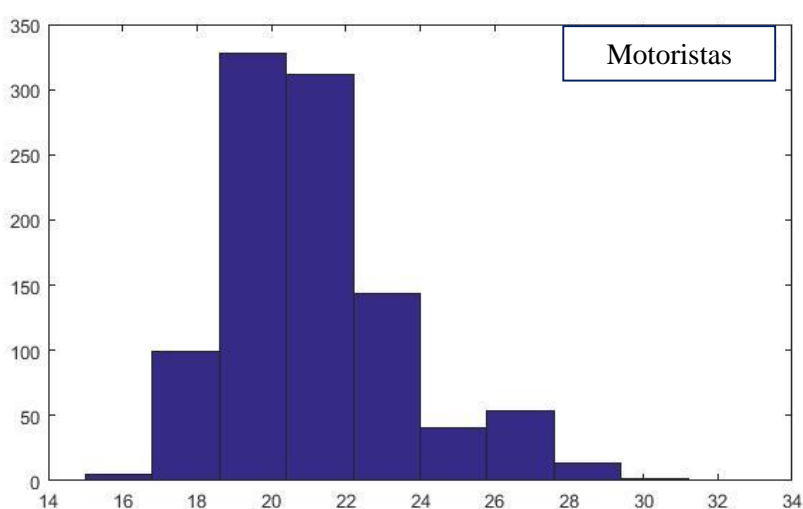
Fonte: Elaboração própria (2016).

O número de motoristas necessário para atender as tarefas ficou baixo em relação ao utilizado pela empresa, provavelmente porque muitas tarefas não foram alocadas.

O modelo tende a criar mais escalas com meia jornada, o que só é favorável para a empresa quando ocorre a troca de um motorista de inteira jornada para um de meia e sem alterar as horas extras pagas.

A Figura 10 apresenta a distribuição do número de motoristas, no eixo horizontal apresenta-se o número de motoristas e no eixo vertical a quantidade de soluções.

Figura 10: Histograma do número de motoristas.

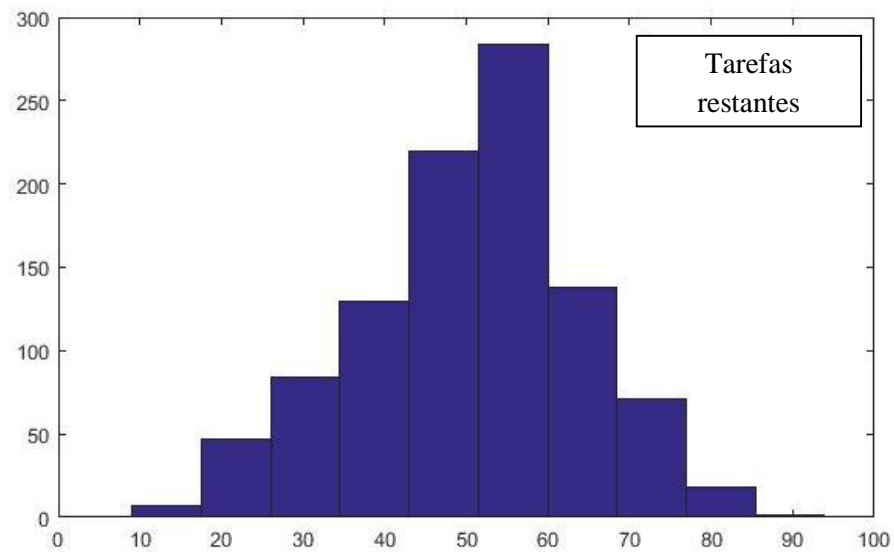


Fonte: Elaboração própria (2016).

O número de tarefas não alocadas apresentou uma média consideravelmente alta, visto que todas deveriam ter sido atribuídas a algum motorista. Para a construção de uma solução inicial há boas soluções, visto que há casos que consegue abordar mais do que 96% das tarefas existentes.

A pior solução deixou de alocar 94 tarefas, o que gerou valores baixos para número de motoristas e tempo ocioso. Na Figura 11 pode ser observada a distribuição das tarefas restantes no eixo horizontal e no eixo vertical a quantidade de soluções.

Figura 11: Histograma das tarefas não alocadas



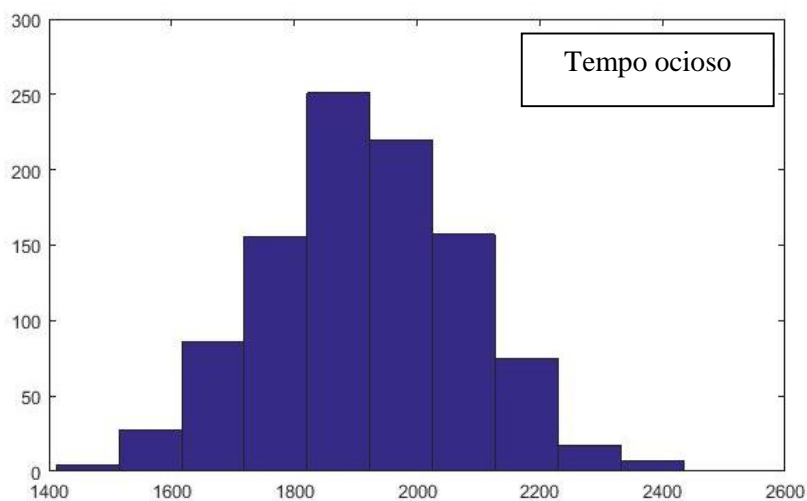
Fonte: Elaboração própria (2016).

O tempo ocioso das soluções apresenta uma média de 1916 minutos, porém em relação à quantidade de tarefas não designadas o tempo não se mostra favorável.

O melhor resultado obteve 1412 minutos de tempo ocioso, e deixou de alocar 83 tarefas. Comparando o ao pior resultado que teve 2433 minutos de tempo ocioso e 41 tarefas não designadas em relação ao tempo utilizado atualmente pela empresa, o método não se mostra qualificado para a construção das escalas, pois a solução da empresa aloca todas as tarefas e tem um tempo de ociosidade de 2381.

Na Figura 12 são apresentados o histograma para o tempo ocioso, tendo no eixo horizontal o total do tempo ocioso e no eixo vertical a quantidade de soluções geradas.

Figura 12: Histograma do tempo ocioso.



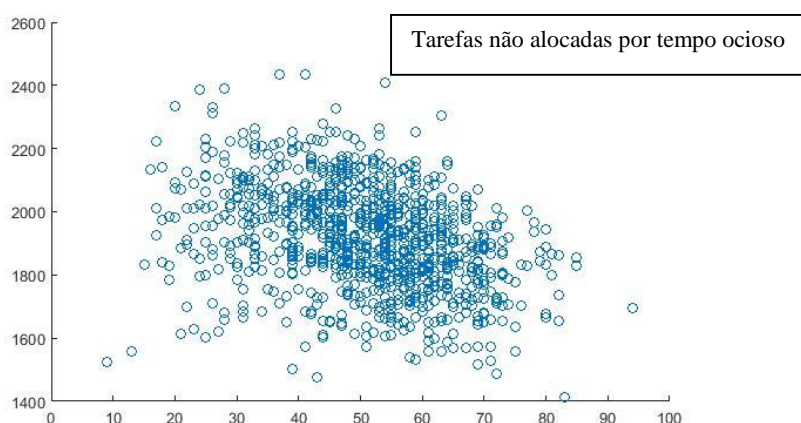
Fonte: Elaboração própria (2016).

A Figura 13 mostra a dispersão dos resultados entre as tarefas que não foram alocadas em relação ao tempo ocioso, sendo representado no eixo horizontal o número de tarefas não alocadas e no eixo vertical o tempo total ocioso das soluções.

Nota-se uma concentração de soluções entre o intervalo de 40 a 70 de tarefas não alocadas em relação ao tempo ocioso na média de 2200 minutos.

Os pontos dispersos no canto direito do gráfico da Figura 13 mostram algumas das boas soluções a serem consideradas para a empresa.

Figura 13: Gráfico de dispersão entre tarefas não alocadas e tempo ocioso.

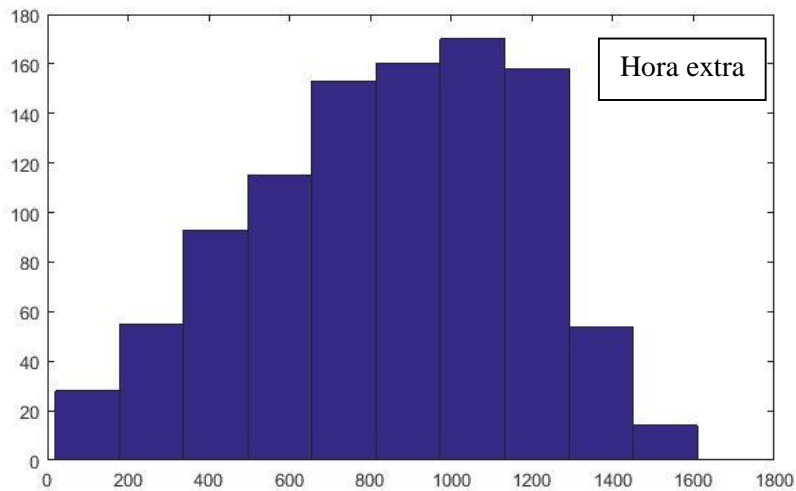


Fonte: Elaboração própria (2016).

O tempo adicionado de horas extras apresentou resultados relativamente altos em relação ao utilizado pela empresa. Provavelmente, porque a implementação tende que os motoristas de inteiras jornadas façam hora extra, visto que o tempo de jornada utilizado na mesma foi considerado o máximo de 09h20min.

Este tempo máximo de jornada foi adotado pois era esperado que as primeiras escalas fizessem um maior número de tarefas, e depois de várias tarefas já terem sido designadas a uma escala as soluções não utilizariam as horas extras. De certa forma, isto ocorreu como será mostrado posteriormente. Na solução apresentada neste estudo dos 43 motoristas apenas 4 trabalham com horas extras, porém, a soma a ser paga triplica em valor financeiro.

Figura 14: Histograma de tempo de hora extra.



Fonte: Elaboração própria (2016).

Como pode ser observado na Figura 14, a maioria das soluções obtidas apresentam altos valores de hora extra a serem pagos. O eixo horizontal do gráfico representa o total de hora extra e o eixo vertical a quantidade de soluções geradas.

A Tabela 6 apresenta os melhores e os piores resultados obtidos em relação aos parâmetros avaliativos.

Tabela 6: Apresentação dos melhores e piores resultados.

No. de motoristas	Tempo ocioso (Min)	Horas extra (Min)	Tarefas não alocadas	Relação ao total de tarefas (%)
33	1412	720	83	24,19
22	1872	20	39	11,37
15	1696	1036	94	27,40
20	1963	1609	50	14,57
30	1524	567	12	3,50
21	2433	1202	41	11,95
33	1559	231	13	3,79

Fonte: Elaboração própria (2016).

A iteração com menor tempo ocioso dos motoristas só poderia ser utilizada como uma solução inicial, visto que 83 tarefas não foram alocadas a algum motorista. Assim seria necessária outra heurística para tornar a solução viável, atribuindo todas as tarefas a algum motorista. Porém ao alocar estas tarefas restantes, o tempo ocioso tenderia a aumentar. Mesmo no pior tempo ocioso de 2433 minutos, 41 tarefas também não foram alocadas.

A solução gerada com o menor número de motoristas, ou seja, com 15 motoristas,

também não pode ser considerada viável por não ter alocado 94 tarefas. Para a solução se tornar viável, ao alocar todas as tarefas restantes o número de motoristas aumentaria.

A melhor solução obtida foi considerada a de menor valor de tarefas não alocadas, com um total de 12 tarefas não alocadas.

Com a intenção de apresentar uma solução sem tarefas restantes foi aplicada uma heurística de melhoria. Quando as restrições de ponto de saída e chegada era respeitada ocorria a adição das tarefas restantes nas escalas já existentes, podendo ser no início ou final de uma escala e ou em algum período de tempo ocioso. Mesmo com a melhoria apenas 6 tarefas foram adicionadas, restando assim 6 tarefas não designadas a um motorista.

A escala obtida após a aplicação da heurística de melhoria é apresentada na Tabela 7.

Tabela 7: Apresentação da escala.

Motorista	Jornada	Tempo ocioso (minutos)	Hora extra (minutos)	Intervalo (minutos)
1	Inteira	77	62	68
2	Meia	63	0	0
3	Meia	53	0	0
4	Inteira	107	33	70
5	Inteira	120	0	67
6	Meia	56	0	0
7	Inteira	120	41	80
8	Inteira	119	0	70
9	Inteira	173	0	58
10	Meia	47	0	0
11	Meia	123	0	0
12	Meia	40	0	0
13	Inteira	49	0	75
14	Inteira	108	0	65
15	Meia	27	0	0
16	Meia	38	0	0
17	Meia	139	0	0
18	Meia	50	0	0
19	Meia	61	0	0
20	Meia	38	0	0
21	Meia	42	0	0
22	Meia	47	0	0
23	Meia	25	0	0
24	Inteira	195	53	70
25	Inteira	148	0	85
26	Meia	49	0	0
27	Meia	99	0	0
28	Inteira	118	0	90
29	Inteira	126	0	93

30	Meia	79	0	0
31	Inteira	136	0	100
32	Meia	44	0	0
33	Meia	94	0	0
34	Inteira	116	0	60
35	Meia	27	0	0
36	Meia	70	0	0
37	Inteira	111	70	75
38	Meia	23	0	0
39	Meia	70	0	0
40	Meia	36	0	0
41	Meia	66	0	0
42	Meia	65	0	0
43	Meia	68	0	0

Fonte: Elaboração própria (2016).

Ao todo foram necessários 43 motoristas, sendo 15 motoristas de inteira jornada e 28 de meia jornada. O tempo ocioso total foi de 3463 minutos e 226 horas extras a serem pagas.

Em comparação com a solução inicial, não houve uma redução no total do tempo ocioso. De fato, ocorreu um aumento de 45,44% em relação ao total de minutos do tempo ocioso e as horas extras a serem pagas triplicaram. A Tabela 7 apresenta os parâmetros e os respectivos resultados.

Tabela 7: Comparação da solução proposta a utilizada na empresa.

Solução	Número de motoristas	Tempo ocioso (Min)	Hora extra (Min)	Restante
Empresa	32	2381	65	0
Proposta	43	3463	226	6

Fonte: Elaboração própria (2016).

As soluções obtidas são consideradas interessantes se forem tratadas como soluções iniciais para se aplicar outro método de melhoria, pois os motoristas de meia jornada apresentam o tempo de ociosidade consideravelmente baixo em relação ao tempo total trabalhado por eles.

Outra sugestão seria construir apenas escalas de meias jornadas através do caminho mínimo, para posteriormente juntar as soluções tentando ao máximo aumentar o número de motoristas com inteira jornada.

Na solução proposta o alto valor do tempo de ociosidade está relacionado ao grande



número de motoristas. Isto pode ter ocorrido devido a alguns fatores de escolha na implementação do método em que as tarefas são selecionadas randomicamente, como exemplo, o vértice inicial e o tempo de jornada antes do intervalo.

Tarefas no horário de pico têm mais possibilidades de sequenciamento com tempos de ociosidade menores, mas se elas não forem escolhidas como vértices iniciais não causam impactos significativos na redução do tempo total ocioso.

Semelhante a isto ocorre com a escolha da tarefa pós-intervalo, visto que o algoritmo escolhe a mais próxima do fim do intervalo, mas havendo uma relaxação a esta restrição, algo de poucos minutos a mais, talvez houvesse outro sequenciamento melhor a ser alocado.

Estas notações podem ser verificadas na matriz de alcançabilidade, visto que estas tarefas apresentam mais posições com valores positivos na matriz do que as demais.

Para cada caso em relação aos parâmetros avaliados pode-se aplicar um método de melhoria, principalmente para a minimização do tempo de hora extra e ao número de motoristas, visto que estes são o que tem alto valor agregado ao custo repassado na passagem.

## 4 CONSIDERAÇÕES FINAIS

O problema de programação da tripulação é de grande importância para o planejamento operacional de empresas do transporte público por representar uma porcentagem significativa dos seus recursos. As escalas de trabalho sendo otimizadas afetam os custos tanto para as empresas de transporte público como para os usuários, pois com economia nos custos salariais pode haver a possibilidade de investimento na qualidade do transporte público e ou também uma redução no valor das tarifas.

A problemática do PPT motivou o presente estudo que teve como objetivo apresentar um método de solução para o mesmo. Após apurar diversas técnicas disponíveis na literatura e trabalhos relacionados que poderiam efetivamente contribuir na resolução do problema, o método de caminhos mínimos foi escolhido. A escolha por este método foi devido a sua implementação computacional ser mais plausível do que aos outros métodos heurísticos.

Este trabalho foi desenvolvido com base nos dados de uma empresa de transporte público na cidade de Joinville. Durante a implementação do estudo, foi possível compreender as características do problema e propor uma metodologia que apresentasse novas soluções. Porém nas novas escalas não foi possível atender a todas as exigências da empresa junto às normas trabalhistas e que ainda reduzissem os custos de operação.

O método sugerido apresenta soluções candidatas a um processo de melhoria e consegue realizar a troca do sequenciamento das tarefas a serem cumpridas com a intenção de minimizar o tempo ocioso entre as mesmas. Acredita-se que aplicando o método proposto junto a outro método heurístico de melhoria, os resultados apresentariam escalas otimizadas com valores menores em relação ao custo. Sugere-se que o método de melhoria seja aplicado visando minimizar o número de motoristas escalados, uma vez que diminuir os custos salariais gerariam maiores economias para a empresa.

A principal dificuldade deste estudo foi referente à etapa de implementação devido ao grande número de restrições matemáticas do problema. As soluções propostas para serem viáveis devem respeitar a todas as restrições apresentadas no capítulo dois. Restrições estas, que na implementação dificultam por sua complexidade computacional nos momentos de construção do sequenciamento de tarefas.

Outra dificuldade no estudo foram as iterações ao longo da implementação, pois tinham um elevado tempo computacional por haver muitas possibilidades de combinação.

Por último, a transcrição do banco de dados para a implementação também foi um fator de dificuldades devido ao seu abundante volume de detalhes à cada tarefa a ser realizada.

Este trabalho mostrasse relevante por propor um modelo de solução a um problema real. Configurado com todas as restrições operacionais e legais, e por auxiliar na disseminação de técnicas de resolução do PPT para o cenário brasileiro, visto que a maioria das empresas ainda geram as escalas de trabalho de forma manual, pois ainda há um número reduzido de softwares capazes de realizar o mesmo.

Tendo economia na etapa de criação de escalas, pode ocorrer um acordo entre empresa e o gestor público em relação a um possível reajuste no valor da tarifa, tornando o uso do transporte público mais acessível à população. Havendo a economia, também poderia ser direcionada para investimentos na qualidade do serviço.

Como proposta para trabalhos futuros sugere-se:

- Aplicar o mesmo método sugerido neste estudo, porém com alguma restrição em relação ao número de motoristas com meia jornada;
- Aplicar o método junto a outro método de melhoria visando minimizar o número de motoristas;
- Expandir a programação para os demais terminais da cidade;
- Realizar um estudo em relação ao custo do tempo ocioso para verificar a real diferença para as empresas;
- A integração da escala de veículos junto a escala dos tripulantes, visto que nem sempre há um ônibus para cada motorista.

## REFERÊNCIAS

ANDRADE, S. **Metaheurística busca tabu para o problema de rodízio de tripulações de ônibus urbanos**. 2013. Dissertação de Mestrado – Programa em Ciência da computação, Universidade Federal de Ouro Preto, Minas Gerais, 2000.

ALVARENGA, B. G. **Um algoritmo híbrido para o problema de roteamento de veículos estático e dinâmico com janela de tempo**. Dissertação de Mestrado – Programa em Ciência da computação, Universidade Federal de Minas Gerais, Minas Gerais, 2005.

BALL, M.; BENOIT-THOMPSON, H. **A lagrangean relaxation based heuristic for the urban transit crew scheduling problem**. Em: Daduna, J. R.; Wren, A. 1998.

BIANCO, L.; BIELLI, M.; MINGOZZI, A; RICCIARDELLI, S.; SPADONI, M. **A heuristic procedure for the crew rostering problem**. European Journal of Operations Research, v. 58, n. 2, 1992.

BOAVENTURA NETO, P. O. **Grafos: teoria, modelos, algoritmos**. São Paulo: Edgard Blucher, 2006.

BUENO, F. **Métodos heurísticos, teoria e implementação**. Instituto federal de Santa Catarina, Araranguá, 2009.

CAPRARA, A.; MONACI, M.; TOTH, P. Models and algorithms for a staff scheduling problem. **Mathematic Programming**, v. 98, 2003.

COLIN, E. **Pesquisa operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas**. Reimpr. Rio de Janeiro, LTC, 2015.

DESROCHERES, M.; SOUMIS, F. A column generation approach to the urban transit crew scheduling problem, **Transportation Science**, 23(1), 1989.

ELIAS, S. **The use of digital computers in the economic scheduling for both man and machine in public transport**, Kansas, EUA: Technical Report, 1964.

EMDEN-WEINERT, T.; PROKSCH, M. Best Practice Simulated Annealing for the Airline Crew Scheduling Problem. **Journal of Heuristics**, 5: 419, 1999.

ERNST, A. T; JIANG, H.; KRISHNAMOORTHY, M. An annotated bibliography of personnel scheduling. **Annals of Operations Research**, 127, 2004.

FISCHETTI, M.; MARTELLO, S.; TOTH, P. **The fixed job schedule problem with spread-time constraints**, Operation Research 35(6), 1987.

FORES, S.; PROLL, L; WREN, A. **An improved ILP system for driver scheduling**. In N. H. M. Wilson (ed), Computer-Aided Transit Scheduling, Spring, 1999.

FRELING, R.; LENTINK, R. M.; WAGELMANS, A.P. **A decision support system for crew planning in passenger transportation using a flexible branch-and-price algorithm**, Annals of Operations Research, 127 (1-4), 2004.

GENDREAU, M.; POTVIN, J. Y. (Ed.) **Handbook of Metaheuristics**. 2. ed. Berlin: Springer, 2010.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & operations research**, 13, 1986.

GOLDBARG, M. C.; LUNA, H. P. L. **Otimização combinatória e programação linear: modelos e algoritmos**. 2. ed. Rio de Janeiro: Campus, 2005.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. **European Journal of Operational Research**, 130, 449-467, 2001.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. 9. ed. São Paulo: Mcgraw Hill, 2013.

IPPUJ. **Joinville Bairro a Bairro**. 2015. Disponível em:<https://ippuj.joinville.sc.gov.br/arquivo/lista/codigo/598-Vers%C3%A3o%2B2015.html>. Acesso em 14 de novembro de 2016.

KIRKPATRICK, S.; GELATT Jr, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. **Science**. 220 (4598), 1983.

LACHTERMACHER, G. **Pesquisa Operacional na tomada de decisões**. 4. ed. São Paulo: Pearson, 2009.

LI, J.; KWAN, R. S. A fuzzy genetic algorithm for driver scheduling. **European Journal of Operation Research**, 147(2), 2003.

LIN, S. Computer solution of the traveling salesman problem. **Bell System Technical Journal**, 1965.

MARINHO, E. H.; OCHI, L. S.; DRUMMOND, L. M.; SOUZA, M. J. F.; SILVA, G. P.: **Busca tabu aplicada ao problema de programação de tripulações de ônibus urbano**. Simpósio Brasileiro de Pesquisa Operacional, XXXVI, 2004.

MAYRINK, V. T. M., SILVA, G. P. Otimização do rodízio de tripulações do sistema de transporte público. **Journal of Transpor Literature**, v. 7, n. 3, 2013.

MESQUITA, M.; MOZ, M.; PAIAS, A.; PAIXÃO, J.; PATO, M.; RESPÍCIO, A. A new model for the integrated vehicle-crew-rostering problem and a computational study on rosters. **Journal of Scheduling**, 14(4), 2011.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers Ops. Research**, 24 (11), 1999.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers Ops. Research**, 24 (11), 1097-1100, 1997.

NURMI, K.; KYNGAS, J.; POST, G. Driver rostering for bus transit companies. **Engineering Letters**, 19 (2), 2011.

PRATA, B. D. A. Programação integrada de veículos e motoristas: uma visão geral. **Revista Eletrônica Sistemas & Gestão**, 2010.

**QualiÔnibus**, 2014. Disponível em:  
<https://www.joinville.sc.gov.br/arquivo/download/codigo/1869-Quali%C3%94nibus%2B-%2Bpesquisa%2Bde%2Bsatisfa%C3%A7%C3%A3o.html>. Acesso em 20 de novembro de 2016

REIS, J. V. A. D. **Heurísticas baseadas em Busca em vizinhança variável para o Problema de programação integrada de veículos e tripulações no transporte coletivo urbano por ônibus**. Dissertação de Mestrado. Escola Politécnica da Universidade de São Paulo, 2008.

REIS, A. F. da S.; E SILVA, G. P. **Um estudo de diferentes métodos de busca e a metaheurística VNS para otimizar a escala de motoristas de ônibus urbano**. Transporte em Transformação XVI - Trabalhos Vencedores do Prêmio CNT - Produção Acadêmica, 2011.

ROCHA, M. K. **Otimização de escalas de serviços de tripulações** – estudo de caso em uma empresa de transporte urbano na região metropolitana de Natal/RN. Dissertação de Mestrado. UFRN, Natal, 2012.

SANTOS, A. G.; MATEUS, G. R. **Crew scheduling urban problem: an exact column generation approach improved by genetic algorithm**. Evolutionary Computation, 2007.

SILVA, G. P. **Uma metodologia baseada na técnica de geração de arcos para o problema de programação de veículos**. São Paulo: Escola Politécnica USP, 2001.

SILVA, G. P.; CUNHA, C. B. Uso da Técnica de Busca em Vizinhança de Grande Porte para a Programação da Escala de Motoristas de Ônibus Urbano. **Transportes**, 2010.

SILVA, G. P.; REIS, A. F. S. A study of different metaheuristics to solve the urban transit crew scheduling problem. **Journal of Transport Literature**, 2014.

SILVA, T.A.; SILVA, G.P. O uso da metaheurística Guide Local Search para resolver o problema de escala de motoristas do ônibus urbano. **Transportes**. v. 23, n. 2, 2015.

SOUZA, D. S. **Uma abordagem híbrida para resolver o problema da escala de motoristas de ônibus urbano**. Dissertação de Mestrado, Programa de pós-graduação em computação, Universidade Federal de Ouro Preto, Minas Gerais, 2014.

STEFANELLO, F. **Hibridização de métodos exatos e heurísticos para resolução de problemas de otimização combinatória**. Dissertação de Mestrado. Programa de pós-graduação em informática, Universidade Federal de Santa Maria, Rio Grande do Sul, 2011.

TAGLIALENHA, S. L. S; ROMERO, R. **Novas Aplicações de Metaheurísticas na Solução do Problema de Planejamento da Expansão do Sistema de Transmissão de Energia Elétrica**. Tese (Doutorado), DEE-FEIS-UNESP, Ilha Solteira, 2008.