



**UNIVERSIDADE FEDERAL DE OURO PRETO
ESCOLA DE MINAS
COLEGIADO DO CURSO DE ENGENHARIA DE
CONTROLE E AUTOMAÇÃO - CECAU**



**RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO DE ÔNIBUS URBANO
UTILIZANDO A METAHEURÍSTICA LNS**

**MONOGRAFIA DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

KLEVISON DANIEL DE OLIVEIRA RIBEIRO

OURO PRETO, 2017

KLEVISON DANIEL DE OLIVEIRA RIBEIRO

**RESOLUÇÃO DO PROBLEMA DE PROGRAMAÇÃO DE ÔNIBUS URBANO
UTILIZANDO A METAHEURÍSTICA LNS**

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Gustavo Peixoto Silva

Ouro Preto
Escola de Minas – UFOP
2017

R484r Ribeiro, Klevison Daniel de Oliveira.
Resolução do Problema de Programação de Ônibus Urbano utilizando a
Metaheurística LNS [manuscrito] / Klevison Daniel de Oliveira Ribeiro. -
2017.

50f.: il.: color; grafs; tabs.

Orientador: Prof. Dr. Gustavo Peixoto Silva.

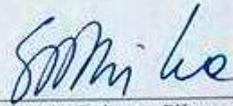
Monografia (Graduação). Universidade Federal de Ouro Preto. Escola de
Minas. Departamento de Engenharia de Controle e Automação e Técnicas
Fundamentais.

1. Ônibus - Programação de sistemas (Computação). 2. Serviço público -
Transporte - Pesquisa Operacional. 3. Serviço publico - Transporte. I. Silva,
Gustavo Peixoto. II. Universidade Federal de Ouro Preto. III. Título.

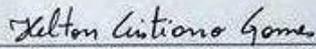
CDU: 681.5

Catálogo: ficha@sisbin.ufop.br

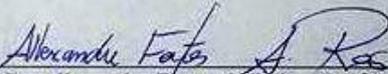
Monografia defendida e aprovada, em 31 de março de 2017, pela comissão avaliadora constituída pelos professores:



Prof. Dr. Gustavo Peixoto Silva - Orientador



Prof. Dr. Helton Cristiano Gomes – Professor Convidado



Prof. M. Sc. Alexandre Fortes S. Reis – Professor Convidado

A minha família que foi o suporte mais importante durante essa jornada.

AGRADECIMENTOS

Primeiramente, agradeço à Deus, por permitir mais uma conquista, pela proteção que me proporcionou durante toda essa jornada e por sempre iluminar meus passos. Agradeço também minha família. Ao Milton, meu pai, à Lílian, minha mãe e também à minha irmã Kethlen por sempre me darem suporte para crescer e me desenvolver mais. Sem eles, essa luta não teria sido vencida.

Não poderia também deixar de agradecer à minha namorada Luísa por todo amor e carinho. Por dividir tantos momentos bons e por estar presente também nos períodos difíceis, confortando e ajudando a superar. Agradeço também aos seus pais Fernando, Patrícia e à toda sua família por todo apoio e auxílio nessa reta final.

Aos amigos e colegas agradeço pelos incentivos, pela ajuda nos estudos e pelos momentos de diversão. Por fim, agradeço à todos professores da UFOP que contribuíram para que esse dia chegasse. Em especial ao professor Paulo, pela ajuda durante o estágio e também ao professor Gustavo, pela oportunidade e por todo conhecimento repassado nesse trabalho.

A todos vocês o meu muito obrigado! Sem vocês eu não teria sucesso em concluir essa jornada!

“O estresse do crescimento é muito menor do que o estresse do fracasso e da estagnação.”

(Flávio Augusto)

RESUMO

O transporte público é um sistema amplamente utilizado pela população, principalmente por aqueles que residem nos grandes perímetros urbanos. Como esse sistema tem um grande impacto no dia-a-dia de milhões de pessoas, ele precisa estar em constante melhora. Uma das etapas da operação do sistema de transporte público é a programação dos veículos que irão realizar as viagens previstas na tabela de horários. Esta alocação pode ser realizada de forma manual pelas empresas, entretanto, o uso de métodos de otimização pode implicar num melhor aproveitamento da frota e uma consequente redução nos custos das empresas. Com base nesse cenário, neste trabalho foi desenvolvida uma versão da heurística *Large Neighborhood Search* (LNS) para resolver o Problema de Programação de Veículos (PPV). Ao solucionar o PPV, pretende-se definir a quantidade mínima de veículos em operação, assim como o detalhamento das atividades de cada veículo em operação. Este detalhamento especifica quais as viagens serão realizadas por quais veículos, o tempo de espera nos terminais e as possíveis viagens ociosas realizadas. A solução para o problema foi construída inicialmente com a busca gulosa, posteriormente foram adaptados métodos de remoção e inserção das viagens aos veículos e finalmente foi implementada a heurística LNS. Foram realizados testes computacionais com dados reais de uma série de empresas que operam na região metropolitana de Belo Horizonte, MG. Os resultados foram comparados com aqueles obtidos por um modelo exato baseado em Algoritmos de Fluxo em Redes. Os resultados se mostraram promissores, visto que na maioria dos casos a diferença entre a solução aproximada e a solução ótima foi pequena.

Palavras-chave: Problema de Programação de Veículos, Pesquisa Operacional, LNS, Programação de Ônibus Urbano, Transporte Público.

ABSTRACT

The Public Transportation System is largely used by people, mostly those who live in the metropolitan areas. Therefore, this process needs to be in constant improvement, because it will impact millions of people daily. One of the steps that make the Public Transportation System is the Vehicle Scheduling that will execute the trips contained in the schedule table. This allocation can be done manually by the companies, however, the use of optimization methods can imply on better use of the fleet and consequently reduce costs. Based on this scenario, in this project was developed one version of the metaheuristic Large Neighborhood Search (LNS) to resolve the Vehicle Scheduling Problem (VSP). To solve the VSP, must be defined the minimum quantity of vehicles, as the detail of activities of the bus used in solution. This details show which trip will be done by which vehicle, the waiting time in the terminals and the idle trips realized. The initial solution on this work was build with basic greedy insertion, after that the removing and insertion heuristics were implemented and finally the LNS was build. Computational tests were made with real data from some companies that operate in Belo Horizonte, MG. The results were compared with one ArcGen model and showed up promising, since in most cases the difference between solutions was little.

Keywords: Vehicle Scheduling Problem, Operational Research, LNS, Urban Bus Scheduling, Public Transportation.

LISTA DE FIGURAS

Figura 1. Processo de Planejamento do Sistema de Transporte Público	22
Figura 2. Pseudocódigo de construção da Solução Inicial	26
Figura 3. Pseudocódigo da Heurística de Remoção Aleatória	27
Figura 4. Pseudocódigo da Heurística de Remoção da Viagem de Pior Custo (versão 1).....	28
Figura 5. Pseudocódigo da Heurística de Remoção da Viagem de Pior Custo (versão 2).....	28
Figura 6. Pseudocódigo da Heurística de Remoção Shaw	30
Figura 7. Pseudocódigo da Heurística de Inserção Gulosa	31
Figura 8. Pseudocódigo da Heurística de Inserção de Arrependimento.....	32
Figura 9. Esquema de funcionamento do LNS.....	32
Figura 10. Pseudocódigo da Metaheurística LNS	33
Figura 11. Desempenho das Heurísticas no Problema SEG-G27	41
Figura 12. Desempenho das Heurísticas no Problema SEG-G48	42
Figura 13. Desempenho das Heurísticas no Problema SEG-G61	42
Figura 14. Desempenho das Heurísticas no Problema SAB-G02	43
Figura 15. Desempenho das Heurísticas no Problema SAB-G27	43
Figura 16. Desempenho das Heurísticas no Problema SAB-G61	44
Figura 17. Desempenho das Heurísticas no Problema DOM-G02.....	44
Figura 18. Desempenho das Heurísticas no Problema DOM-G27.....	45
Figura 19. Desempenho das Heurísticas no Problema DOM-G61.....	45

LISTA DE TABELAS

Tabela 1. Entradas e saídas de cada subproblema do Planejamento do Sistema de Transporte Público	24
Tabela 2. Resultados dos testes do LNS para os problemas de segunda-feira	35
Tabela 3. Resultados dos testes do LNS para os problemas de sábado	36
Tabela 4. Resultados dos testes do LNS para os problemas de domingo	37
Tabela 5. Resultados dos testes do LNS para o problema SEG-G27	38
Tabela 6. Resultados dos testes do LNS para o problema SEG-G48	38
Tabela 7. Resultados dos testes do LNS para o problema SEG-G61	38
Tabela 8. Resultados dos testes do LNS para o problema SAB-G02	39
Tabela 9. Resultados dos testes do LNS para o problema SAB-G27	39
Tabela 10. Resultados dos testes do LNS para o problema SAB-G61	40
Tabela 11. Resultados dos testes do LNS para o problema DOM-G02	40
Tabela 12. Resultados dos testes do LNS para o problema DOM-G27	40
Tabela 13. Resultados dos testes do LNS para o problema DOM-G61	41
Tabela 14. Número de veículos utilizados para cada solução	46

LISTA DE SIGLAS

ALNS	<i>Adaptive Large Neighborhood Search</i>
FO	Função Objetivo
LNS	<i>Large Neighborhood Search</i>
PPT	Problema de Programação das Tripulações
PPV	Problema de Programação dos Veículos
STP	Sistema de Transporte Público
TH	Tabela de Horários

SUMÁRIO

1. INTRODUÇÃO	14
1.1. Justificativas	14
1.2. Objetivos.....	15
1.3. Metodologia.....	15
1.4. Estrutura do Trabalho	16
2. REVISÃO DA LITERATURA	17
3. MÉTODO DE RESOLUÇÃO DO PROBLEMA	22
3.1. O Problema de Programação de Veículos	22
3.2. Função Objetivo	25
3.3. Solução Inicial	26
3.4. Heurísticas de Remoção de Viagens	26
3.4.1. Remoção Aleatória	27
3.4.2. Remoção da Viagem de Pior Custo.....	27
3.4.3. Remoção Shaw	29
3.5. Heurísticas de Inserção de Viagens	30
3.5.1. Inserção Gulosa	30
3.5.2. Inserção de Arrependimento.....	31
3.6. Algoritmo LNS Para o PPV.....	32
4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS	34
5. CONCLUSÃO.....	47

1. INTRODUÇÃO

1.1. Justificativa

O Sistema de Transporte Público – STP é um dos serviços coletivos mais importantes e utilizados pela população nas cidades de médio e grande porte. Uma vez que se observa o seu grande impacto no cotidiano das pessoas, torna-se necessário utilizar as ferramentas computacionais disponíveis para definir um planejamento otimizado das operações, com a finalidade de torná-lo mais eficiente e econômico para todos. Melhorar a eficiência desse serviço pode trazer economia para as empresas e com menores custos de operação, as tarifas não precisam de aumentos recorrentes, beneficiando também o público que utiliza o sistema. Com isso, se percebe a relevância de se buscar soluções que melhorem o serviço e ao mesmo tempo otimizem o sistema de forma a diminuir os custos e aumentar a qualidade do transporte público. Além disso, otimizar a frota é um fato de grande importância para o contexto atual, pois também pode reduzir a emissão de gases poluentes, contribuindo assim para a preservação do meio ambiente.

Planejar as operações de um STP é uma tarefa de grande complexidade e por isso mesmo é um desafio chegar a uma definição que seja eficiente e atenda razoavelmente os interesses dos diversos agentes envolvidos. Devido à sua complexidade, o planejamento é dividido em várias etapas, sendo que a solução de uma etapa serve como dado de entrada para a etapa subsequente. Segundo Silva (2001), as principais etapas deste processo são:

- *Definição das rotas:* Nesta fase uma pesquisa é realizada junto à população para entender como funciona o fluxo de pessoas dentro da região em que se está planejando o serviço. Com base em uma pesquisa do tipo “origem-destino” do fluxo de pessoas no município, são definidas as rotas, ou linhas por onde devem circular os ônibus.
- *Definição da tabela de horários:* com base na demanda por viagem em cada linha, definida na etapa anterior, é gerada uma tabela de horários para cada linha. Nesta tabela é estipulada a frequência com que as viagens serão executadas. Geralmente, a demanda sobre as viagens cresce em três horários distintos: *i*) na parte da manhã, com

origem no bairro e o centro como destino, *ii*) no horário do almoço e *iii*) no fim da tarde, no sentido centro-bairro.

- *Programação dos veículos*: nesta etapa é definida a quantidade de veículos necessária para realizar todas as viagens presentes na Tabela de Horários - TH. Além disso, também é gerado o *bloco do veículo*, que contém, para cada veículo, as viagens que ele deve executar e os tempos de parada entre cada viagem, além das viagens ociosas que por ventura tenha que fazer, como por exemplo, a partida e o retorno à garagem.
- *Programação e rodízio das Tripulações*: após a programação dos veículos, o passo seguinte consiste em definir os turnos diários das tripulações. Esses turnos devem ser estipulados respeitando diversas leis trabalhistas como: tempo total de trabalho, tempo mínimo de descanso, entre outros. Ao final dessa fase ainda deve ser realizado o rodízio das tripulações, definindo assim a escala de trabalho mensal de cada tripulação.

Neste trabalho é abordada a terceira etapa desse processo, ou seja, a resolução do Problema de Programação dos Veículos – PPV, onde deve ser definida a quantidade de veículos necessária para realizar todas as viagens estabelecidas na etapa anterior e também o detalhamento das viagens a serem realizadas por cada veículo, de tal forma que todas as viagens da tabela de horários sejam executadas. Toda a programação é realizada com o objetivo de obter os menores custos operacionais e o menor número de veículos possível para realizar as viagens presentes na tabela de horários.

1.2. Objetivos

O objetivo deste trabalho consiste em implementar a heurística LNS para resolver, de forma otimizada, o PPV.

1.3. Metodologia

Para resolver o PPV foi implementada a heurística *Large Neighborhood Search* – LNS, que é um método que tenta melhorar uma dada solução por meio de uma abordagem do tipo “destrói-reconstrói” (SCHRIMPF *et al.*, 2000). Abordagens deste tipo destroem parte da

solução corrente e tentam reconstruí-la por meio de procedimentos que obtenham uma nova solução melhor do que as soluções anteriores.

A heurística implementada foi testada com dados de empresas que operam na região metropolitana de Belo Horizonte e seus resultados foram comparados com aqueles obtidos por métodos exatos de solução. A importância de se implementar uma heurística eficiente para o PPV se deve à limitação dos métodos exatos em representar características impossíveis de se incluir no modelo exato, como por exemplo a necessidade de controlar o tempo máximo de operação de um veículo, controlar o número de trocas de linhas realizadas pelos veículos, entre outras.

1.4. Estrutura do Trabalho

Neste trabalho os conteúdos estão dispostos da seguinte forma: no Capítulo 1 temos uma breve introdução sobre o contexto que cerca o problema. Justificativas relevantes, objetivos e metodologia adotada também estão presentes nesta seção. No Capítulo 2 está disposto o levantamento bibliográfico realizado para o suporte deste trabalho.

No Capítulo 3 são apresentados os métodos utilizados para resolver o PPV. A metaheurística LNS é descrita detalhadamente nesta parte do trabalho, assim como a construção da solução inicial e as heurísticas de remoção e inserção utilizadas para otimizar a solução.

Para descobrir a eficiência dos métodos implementados para solução do PPV, foram realizados diversos testes em vários problemas diferentes. Os resultados obtidos por esses métodos estão apresentados no Capítulo 4. Por fim, no Capítulo 5 tem-se a conclusão do que foi estudado neste trabalho.

2. REVISÃO DA LITERATURA

Para iniciar a implementação da heurística de resolução do problema, foi necessário fazer um estudo sobre o assunto. Inicialmente procurou-se entender o problema em si e suas dificuldades, juntamente com as técnicas apresentadas na literatura que foram utilizadas para encontrar boas soluções.

No trabalho de Pepin *et al.* (2009) é feita uma comparação entre cinco heurísticas diferentes para a solução do Problema de Programação de Veículos com Múltiplos Depósitos, ou seja, os veículos partem de várias garagens diferentes para realizar uma dada tabela de horários. Neste caso, o PPV não foi utilizado sobre o sistema de transporte público e sim sobre um ponto de vista de logística comercial, onde temos tarefas que são executadas apenas uma vez por um único veículo. Um exemplo de aplicação desse tipo de problema é a entrega de mercadorias, onde um produto ou compra, por exemplo, é entregue ao consumidor uma única vez sendo transportado por um único veículo. As cinco heurísticas implementadas neste trabalho foram: *Branch and Cut* truncado, heurística Lagrangeana, Geração de Colunas truncado, *Large Neighborhood Search* (LNS) e Busca Tabu. O LNS utiliza o conceito já citado aqui de destruir e reconstruir a solução de forma a otimizá-la. Três remoções foram adotadas nesse estudo: *i*) aleatória; *ii*) menos frequente e *iii*) mais próximo. A remoção aleatória retira da solução uma programação via sorteio. Já a menos frequente retira da solução aquelas programações que aparecem pouco nas soluções das iterações anteriores. A retirada do mais próximo remove da solução programações similares.

O termo programação aqui citado representa um conjunto de viagens possíveis de serem realizados por um veículo, e nos métodos exatos, cada programação é representada por uma coluna na matriz de restrições do problema. Para reconstruir a solução, o algoritmo emprega a técnica de geração de colunas para o problema (BARNHART *et al.*; 1998). Os autores concluem que o método mais eficiente com pequenos tempos de processamento é o LNS. Ou seja, é a heurística que produz o melhor resultado com o menor tempo de resposta. Quando se tem disponível maior tempo no processamento, a melhor alternativa encontrada pelo autor foi o método de Geração de Colunas.

Outro trabalho que foi utilizado como base para essa monografia foi o de Ropke e Pisinger (2006). Neste trabalho os autores propõem as heurísticas LNS e ALNS (*Adaptive Large Neighborhood Search*), que é uma adaptação do LNS, para resolver um problema de coleta e entrega com janela de tempo (*pickup and delivery problem with time windows*). No LNS apenas uma heurística é usada para destruir uma solução e outra para reconstruir, enquanto que no ALNS são utilizadas várias heurísticas para remoção e inserção, de forma que as heurísticas mais bem sucedidas são utilizadas mais vezes. Para o problema de coleta e entrega devem ser estabelecidas rotas para atender um conjunto de pedidos, sendo que cada pedido consiste em coletar um produto em um determinado local e entregá-lo em outro local dentro de uma janela de tempo previamente especificada. No caso, tanto a coleta quanto a entrega devem ocorrer dentro de uma determinada faixa de tempo.

A metaheurística ALNS utilizada nesta referência adota três procedimentos de remoção e dois de inserção dos pedidos. As remoções escolhidas pelo autor foram: *i*) por similaridade, *ii*) aleatória e *iii*) do pior pedido. Na remoção por similaridade são removidos os pedidos que tem características semelhantes, como tempo de entrega, localidade, amplitude da janela de tempo. A partir de um primeiro pedido removido aleatoriamente, são retirados outros pedidos similares, segundo a métrica estabelecida e considerando uma dada aleatoriedade. A remoção aleatória retira pedidos da solução sorteados aleatoriamente. Já no método de retirada do pior pedido, são analisados todos os pedidos incluídos na programação. Os pedidos que produzem os maiores custos na solução final são retirados considerando uma dada aleatoriedade. Quanto às inserções, são propostas: *i*) inserção gulosa e *ii*) inserção de arrependimento. Na inserção gulosa são reinseridos na solução primeiro os pedidos com menor custo de inserção, na melhor posição naquele momento. Já a heurística de arrependimento tenta prever possíveis inserções caras futuramente para evitar o arrependimento pela inserção dos primeiros pedidos. Neste procedimento, cada pedido fora da solução é colocado na sua primeira e na sua segunda melhor posição. O pedido que tiver a maior diferença entre a segunda e a primeira melhor posição deve ser inserido antes. Ela procura identificar possíveis desvantagens em inserir um pedido diretamente na sua melhor posição, uma vez que essa inserção prematura pode implicar um custo mais elevado pela obrigatoriedade de inserir o pedido na sua segunda melhor posição na solução.

O método ALNS, atribui uma pontuação para as heurísticas que produzem os melhores resultados. Assim, quanto mais pontuado for o par de heurísticas (remoção, inserção), maiores

serão as chances de cada heurística do par ser utilizada no decorrer das iterações da heurística. Ao final, os autores concluem que é mais vantajoso o uso de concorrência entre heurísticas do que se ter apenas uma remoção e uma inserção, recomendando o uso e atestando a robustez do método ALNS para o problema.

A metaheurística ALNS foi utilizada também por Martins (2017). Neste caso, o método foi usado para resolver o Problema de Programação de Tripulações, que é uma fase do Planejamento do Transporte Público posterior ao Problema de Programação dos Veículos. Assim como no trabalho de Ropke e Pisinger (2006), nesse estudo também estão presentes as heurísticas de inserção gulosa e de arrependimento para reconstruir uma solução destruída pelas remoções. As heurísticas de remoção também foram as mesmas usadas na citação anterior: randômica, pior custo e similaridade. Além das três, foi também utilizada a remoção média. Nessa heurística de remoção as jornadas que ultrapassam o número médio de tarefas tem o excedente removido de forma que essas tarefas retiradas sejam realocadas em outras jornadas. Os testes nesse trabalho foram realizados utilizando dados reais de várias empresas que atuam na região metropolitana de Belo Horizonte, MG. Os resultados obtidos foram satisfatórios, uma vez que superaram alguns métodos encontrados na literatura e também as soluções adotadas pelas empresas.

Silva e Gualda (2002) abordam o PPV utilizando um algoritmo de Geração de Arcos - *ArcGen*. Nesse artigo é abordado o sistema de transporte público da cidade de Reading, no Reino Unido e Sorocaba, no estado de São Paulo. O método *ArcGen* consistem em aplicar um modelo de fluxo em redes do tipo pseudo-assinalamento para representar o problema. O modelo reduz o número de arcos da rede considerando apenas os arcos curtos, ou seja, desconsiderando as ligações que representam uma dupla pegada para o veículo, denominados arcos longos da rede. A partir da primeira solução ótima, é verificada a existência de arcos longos que podem contribuir com a melhoria da solução, por meio dos valores duais dos nós da rede. Quando nenhum arco longo puder melhorar a solução, esta será a solução ótima. O *ArcGen* foi comparado com a heurística chamada BOOST, desenvolvida por Kwan e Rahin (1999). Os autores detectam uma pequena vantagem do primeiro método, sendo que em alguns testes a heurística não atingiu a solução ótima.

Outros trabalhos também foram estudados para o desenvolvimento desta monografia, porém tiveram menos peso na produção do mesmo. Kliewer et al. (2011) resolvem de forma

integrada o Problema de Programação de Veículos juntamente com o Problema de Programação de Tripulação - PPT. Tradicionalmente, estas etapas são realizadas sequencialmente determinando a Programação dos Veículos antes da Programação das Tripulações. Uma possibilidade para melhorar a eficiência da programação, em relação ao planejamento sequencial, é considerar a programação de veículos e das tripulações simultaneamente. Uma abordagem simultânea aumenta a flexibilidade para Programação das Tripulações. Outra forma de melhorar a eficiência é aumentar a flexibilidade enquanto resolve o PPV. Uma TH, definida com base na demanda do cliente, estabelece o conjunto de viagens utilizadas para transportar os passageiros. Estas viagens servem como entrada para resolver o PPV. Geralmente, presume-se que o ponto inicial e final de cada viagem é fixo, assim como seus horários de início e de término. No entanto, se a TH for ligeiramente alterada, modificando um pequeno número de viagens dentro de uma pequena janela de tempo, o espaço de soluções factíveis aumenta drasticamente, permitindo um grau de liberdade adicional no PPV. Neste trabalho é utilizada uma combinação destes dois conceitos – Programação Integrada Veículo-Tripulação e Janelas de Tempo para as viagens. Sendo investigado e comparado variantes da janela de tempo no processo de planejamento do sistema. O modelo de rede subjacente à abordagem se baseia em uma estrutura de rede tempo-espaço. O problema integrado é resolvido com a técnica de Geração de Coluna combinada com Relaxação Lagrangeana. Para reduzir os tempos computacionais, é determinado heurísticamente um subconjunto de viagens críticas para os custos da operação. O deslocamento destas viagens pode levar a uma diminuição no número de veículos e de tripulações e/ou nos custos operacionais.

Ming *et al.*. (2012) resolvem o PPV relacionado à programação de ônibus regional cujos tempos de viagem não são determinístico, mas estocásticos. Os autores resolvem o PPV para múltiplas garagens com tempo de viagem estocástico, por meio de um algoritmo do tipo Colônia de Formigas.

Ibarra-Rojas *et al.*. (2014), por sua vez, trabalha com o PPV integrado à definição da TH, analisando o *trade-off* entre o nível de serviço e os custos operacionais de sistemas de transporte. No problema de definição da TH, deve-se maximizar o número de passageiros beneficiados por transferências bem sincronizadas, enquanto no problema PPV busca-se minimizar os custos operacionais, que estão relacionados ao tamanho da frota. Neste artigo são apresentados dois modelos de Programação Linear Inteira, um para cada problema.

Posteriormente, os autores apresentam um método ε -*constraint* para resolver conjuntamente a definição da TH e o PPV como um problema bi-objetivo. Isso permite analisar a relação de troca entre estes dois critérios em termos de fronteira de Pareto. Experimentos numéricos mostram que a abordagem proposta pode resolver cenários com até 50 linhas de ônibus.

Com esta revisão, foi observado que embora o PPV seja um problema clássico, ele é pouco estudado e aplicado à realidade brasileira. Desta forma, torna-se importante desenvolver trabalhos que abordam a aplicação de modelos de otimização à nossa realidade.

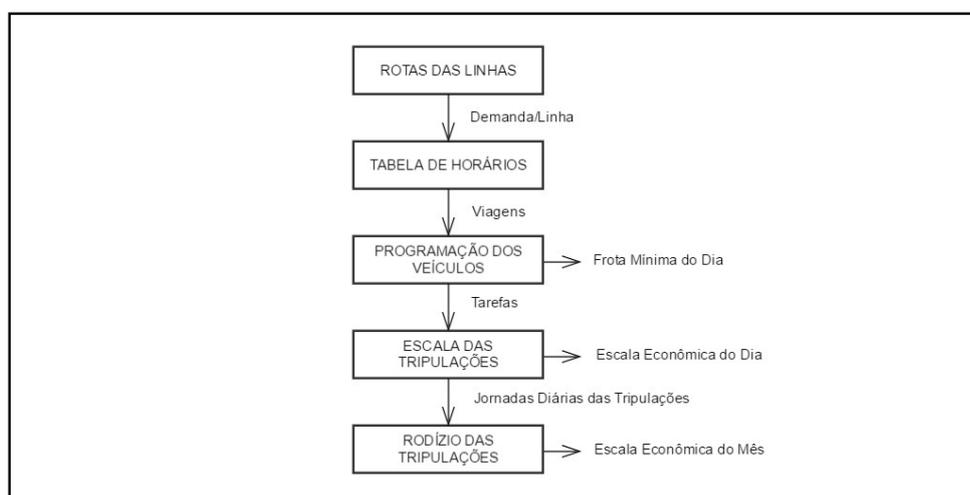
3. MÉTODO DE RESOLUÇÃO DO PROBLEMA

Inicialmente, para se apresentar o método implementado para resolver o PPV, é necessário mostrar de forma mais detalhada as características do problema. A seguir são apresentados os detalhes do problema estudado, da modelagem e da implementação da metaheurística LNS desenvolvida para resolver o PPV.

3.1. O Problema de Programação de Veículos

O Problema de Programação de Veículos nos sistemas de transporte público é apenas uma das etapas que compõem o planejamento desse sistema. Como o processo é muito complexo, é feita uma subdivisão dos problemas para facilitar o planejamento da operação do sistema. Esses subproblemas, que compõem o Sistema de Planejamento do Transporte Público, começam com a definição de rotas, a criação da tabela de horários de viagens, a programação de veículos, a programação de tripulações e o rodízio das tripulações. O esquema dessa subdivisão pode ser visto na Figura 1.

Figura 1. Processo de Planejamento do Sistema de Transporte Público



A definição das rotas, ou linhas, é a primeira etapa do processo. Nessa etapa faz-se um estudo das necessidades da população em se locomover. As rotas são traçadas de forma que sejam atendidos os pontos com maiores demandas de passageiros em um único traçado. Ao longo desses caminhos são determinadas as estações e pontos de parada dos veículos. Para definir de forma adequada as rotas de uma localidade, deve ser realizado um estudo da “Matriz de Origem e Destino dos Usuários do Transporte Coletivo”. Ou seja, deve-se conhecer a

demanda por viagem existente em cada período do dia, e a infra-estrutura disponível para que as rotas sejam traçadas. Existem critérios e modelos de otimização muito bem definidos para que seja definida uma rede de transporte público eficiente. Para maiores detalhes são recomendados os trabalhos de Ceder e Wilson (1986) e Pattnaik *et al.* (1998).

A segunda fase do processo de planejamento do sistema de transporte público é a criação da tabela de horário das viagens que serão realizadas. Nessa etapa é analisado o volume de passageiros que precisam ser transportado em cada rota. Dessa forma, é definido o número de viagens executadas em cada rota para um dado intervalo de tempo, ou seja, a frequência de partida dos veículos do ponto inicial de cada rota.

Definida a TH, inicia-se a programação dos veículos, que consiste em encontrar o número mínimo de veículos necessários para executar todas as viagens da TH, e a alocação de cada viagem a um dado veículo em operação. Com a programação de veículos definida, parte-se para a programação de tripulações (dupla motorista e cobrador). Nela, são definidas as jornadas diárias de trabalho das tripulações, respeitando a legislação definida e também as políticas da empresa. Uma vez definidas as jornadas diárias, deve-se estabelecer o rodízio das tripulações ao longo do mês, ou seja, definir os dias de trabalho e de descanso de cada motorista e cobrador da empresa, assim como as jornadas a serem executadas ao longo do mês por cada dupla. Além de minimizar o total de tripulação necessária, nesta etapa pretende-se também minimizar o total de horas extras pagas pela empresa ao final do mês.

Este trabalho tem como foco a resolução do Problema de Programação de Veículos – PPV, que tem como dados de entrada a TH com as informações de todas as viagens a serem realizadas pela empresa em um dado tipo de dia. A TH realizada nos dias úteis é diferente da TH do sábado, que por sua vez é diferente da TH realizada aos domingos. Para um dado tipo de dia, as informações disponíveis para cada viagem são: *i*) um identificador da viagem, *ii*) o seu horário de início, *iii*) o horário de fim, *iv*) o local de partida e *v*) local de chegada da viagem. Deve ser informada também a matriz de distância entre todos os pontos de início e término das viagens e a distância destes pontos até a garagem. As regras operacionais da empresa também fazem parte das informações necessárias para modelar e resolver o PPV. Como solução do PPV, tem-se o número de veículos na frota e o detalhamento das viagens que cada veículo deve executar ao longo do dia, inclusive com os deslocamentos ociosos realizados pelos veículos, de tal forma que cada viagem da TH seja realizada uma única vez.

Este detalhamento das viagens de cada veículo é dito *bloco de viagens* do veículo ou simplesmente *bloco do veículo*. O objetivo é executar todas as viagens de forma que o menor número de veículos seja usado, gerando o menor custo com deslocamentos ociosos e tempos de espera nos terminais.

Uma programação que representa uma solução do PPV deve obedecer algumas restrições inerentes ao problema. Primeiramente, deve-se considerar a possibilidade de haver Dupla Pegada, que ocorre quando um veículo fica ocioso entre uma viagem e outra por um tempo superior a duas horas. Além disso, existe uma limitação referente ao número de trocas de linha que cada veículo pode realizar. A troca de linha ocorre sempre que um veículo faz uma viagem sem passageiros entre o término de uma viagem e início da próxima. Um veículo também não pode exceder 23 horas de operação, devendo permanecer pelo menos uma hora por dia na garagem para limpeza, abastecimento e pequenos reparos. Deve ser respeitado também um tempo de embarque para os passageiros. Dessa forma, uma viagem só pode ser realizada por um veículo, se o horário de término da viagem anterior mais o tempo de embarque definido para a viagem que se deseja realizar for menor do que o horário inicial da mesma. Além disso, o tempo de deslocamento entre o ponto final da viagem anterior até o ponto inicial da viagem que se deseja realizar também deve ser considerado.

A Tabela 1 mostra as etapas do Planejamento do Sistema de Transporte Público. É interessante notar que a entrada de cada uma dessas etapas é a saída da fase anterior. As entradas da TH são as rotas definidas no processo anterior. As tabelas de horários servem de entrada para a programação de veículos. A definição do bloco de veículos serve para definir a programação das tripulações, e ela, por sua vez, funciona como entrada para o rodízio das tripulações.

Tabela 1 – Entradas e saídas de cada subproblema do Planejamento do Sistema de Transporte Público (continua)

SISTEMA DE TRANSPORTE PÚBLICO		
PROBLEMA	ENTRADA	SAÍDA
<i>Definição de Rotas</i>	Pesquisa sobre necessidades da população	Rotas que serão percorridos pelos veículos
<i>Tabela de Horários</i>	Rotas definidas	Horário de partida das viagens

Tabela 1 – Entradas e saídas de cada subproblema do Planejamento do Sistema de Transporte Público (conclusão)

<i>Programação de Veículos</i>	Tabela com os horários das viagens	Bloco de veículos com todas as viagens que devem ser realizadas por cada carro.
<i>Programação da Tripulação</i>	Bloco de Veículos	Jornada diária da tripulação
<i>Rodízio da Tripulação</i>	Jornada Diária da Tripulação	Jornada Mensal da tripulação

3.2. Função Objetivo

Para se calcular o custo de uma solução do PPV, são utilizados os seguintes parâmetros:

- i)* o custo fixo, caracterizado pelo custo de cada veículo utilizado na frota;
- ii)* o tempo total de viagens ociosas, realizadas entre a garagem e os pontos iniciais e finais das linhas, assim como o deslocamento entre os pontos finais e iniciais das linhas, quando há a troca de linhas durante a operação de um veículo. Ela é caracterizada pelo traslado do veículo sem passageiros.
- iii)* O tempo total de espera nos terminais, quando o veículo permanece parado esperando para realizar a próxima viagem.

A Função Objetivo - FO a ser minimizada no PPV é dada na expressão (1). Essa equação envolve o custo fixo e os custos variáveis em uma função linear.

$$\text{Custo da Solução} = \alpha \times N^{\circ} \text{ de Veículos} + \beta \times \text{Viagem Ociosa} + \gamma \times \text{Tempo de Espera} \quad (1)$$

O parâmetro α representa o custo fixo para cada veículo utilizado na solução. Neste caso, $\alpha = 1.000$. O parâmetro β atribui o peso ao tempo total de viagem ociosa e seu valor é igual a 2. O parâmetro γ , por sua vez, representa o peso do tempo total de espera nos terminais que possui peso 1. Atribuindo valores aos parâmetros da FO é possível calcular o custo de uma solução qualquer, visto que é conhecido o número de veículos, o tempo total de viagem ociosa e de espera nos terminais.

3.3. Solução Inicial

A solução inicial foi gerada utilizando um algoritmo guloso. Inicialmente as viagens são armazenadas em uma *Lista* em ordem crescente pelos seus horários de início. A construção adotou o seguinte critério: inicializar o primeiro veículo e torná-lo o veículo corrente. Alocar ao veículo corrente a primeira viagem da *Lista*. Posteriormente, percorrer a *Lista* e atribuir ao veículo corrente a primeira viagem que tiver seu início no ponto final da última viagem alocada ao veículo e cujo horário de início seja maior ou igual ao horário de término da última viagem alocada. Quando ao veículo corrente não puder ser alocada nenhuma viagem na *Lista*, inicializar um novo veículo e repetir o procedimento. O pseudocódigo desse procedimento pode ser visto na Figura 2.

Figura 2 – Pseudocódigo da Construção da Solução Inicial

SOLUÇÃO INICIAL ($vg \in \text{Viagens}$, $vc \in \text{Veículos}$)
<pre> n = 1; j = 1; 1. Enquanto (n ≤ total de viagens){ 2. Para(vg = 1; vg ≤ total de viagens; vg++){ 3. Se (inserção de vg em vc_j for possível){ 4. insira vg na lista de viagens de vc_j; 5. n = n + 1; 6. }Fim_Se 7. }Fim_Para 8. j = j + 1; 9. }Fim_Enquanto </pre>

A partir da solução inicial, a minimização do custo foi realizada utilizando o método LNS. Para isso foi necessário implementar várias heurísticas de remoção e inserção, para destruir e reconstruir uma nova solução potencialmente com melhor custo.

3.4. Heurísticas de Remoção de Viagens

As heurísticas de remoção adotam alguns critérios para retirar viagens da solução. As viagens removidas são armazenadas em um *banco de viagens* onde permanecem até que sejam realocadas pelas heurísticas de inserção. Os métodos de inserção, por sua vez, retiram essas

viagens do *banco de viagens* e inserem na melhor posição possível, de acordo com o critério adotado na sua definição. As heurísticas de remoção adotadas foram: aleatória, pior posição e similaridade, as quais são detalhadas a seguir.

3.4.1. Remoção Aleatória

Na remoção aleatória, é sorteado um número no intervalo $[1, \text{total de viagens}]$. O número sorteado corresponde à viagem a ser retirada da solução e enviada ao *banco*. O sorteio é feito até que o número desejado de remoções q seja alcançado, sem que haja repetição nas viagens sorteadas. Na Figura 3 pode ser visto o pseudocódigo dessa heurística de remoção.

Figura 3 – Pseudocódigo da Heurística de Remoção Aleatória

REMOÇÃO ALEATÓRIA ($q \in \mathbb{N}$, $s \in \text{soluções}$)
<p>$L = \text{lista de viagens removidas de } s;$ $t = \text{total de viagens da solução } s;$</p> <ol style="list-style-type: none"> 1. Enquanto ($L < q$) { 2. $L = \text{recebe uma viagem sorteada no intervalo } [1, t];$ 3. $\text{retira viagem sorteada da solução } s;$ 4. } Fim_Enquanto

Fonte: Adaptado de Ropke e Pisinger (2006)

3.4.2. Remoção da Viagem de Pior Custo

A Remoção da Viagem de Pior Custo, para cada iteração, retira uma viagem da solução corrente e calcula o valor da FO sem a mesma, armazenando a variação do custo produzida por cada uma delas. Ao final, são retiradas viagens i que maximizam a diferença $\Delta_i = \text{FO}(\text{solução}) - \text{FO}(\text{solução} - \{i\})$. Para essa remoção foram criadas duas versões, apresentadas nas Figuras 4 e 5 respectivamente. Os parâmetros L , q , n e t apresentados na versão 1 são os mesmos que constam na versão 2. A primeira versão é uma adaptação da implementação original, apresentada por Ropke e Pisinger (2006), a qual está representada na segunda versão.

Figura 4 – Pseudocódigo da Remoção da Viagem de Pior Custo (versão 1)

REMOÇÃO DA VIAGEM DE PIOR CUSTO ($q \in \mathbb{N}$, $s \in \text{soluções}$, p_{pior})
VERSÃO 1
<p>$n = 0$; $t = \text{total de viagens da solução } s$;</p> <ol style="list-style-type: none"> 1. <u>$L = \text{lista de custos da solução } s \text{ retirando-se cada viagem}$</u>; 2. <u>Ordena L (em ordem decrescente de custo)</u>; 3. Enquanto ($n < q$) { 4. sorteia um número y no intervalo $[0, 1)$; 5. $r = L[y^{p_{\text{pior}}} L]$; 6. remove r da solução s e insere no banco de viagens; 7. $n = n + 1$; 8. } Fim_Enquanto

Fonte: Adaptado de Ropke e Pisinger (2006)

Figura 5 – Pseudocódigo da Remoção da Viagem de Pior Custo (versão 2)

REMOÇÃO DA VIAGEM DE PIOR CUSTO ($q \in \mathbb{N}$, $s \in \text{soluções}$, p_{pior})
VERSÃO 2
<p>$n = 0$; $t = \text{total de viagens da solução } s$;</p> <ol style="list-style-type: none"> 1. Enquanto ($n < q$) { 2. <u>$L = \text{lista de custos da solução } s \text{ retirando-se cada viagem}$</u>; 3. <u>Ordena L (em ordem decrescente de custo)</u>; 4. sorteia um número y no intervalo $[0, 1)$; 5. $r = L[y^{p_{\text{pior}}} L]$; 6. remove r da solução s e insere no banco de viagens; 7. $n = n + 1$; 8. } Fim_Enquanto

Fonte: Adaptado de Ropke e Pisinger (2006)

A diferença entre as duas versões está no fato de que na versão 1 o vetor L é calculado apenas uma vez (linhas 1 e 2), enquanto na versão 2 o vetor L é recalculado após a retirada de cada viagem (linhas 2 e 3). Se por um lado, na versão 1 é realizado um número menor de cálculos, na versão 2 as informações são mais precisas ao recalculer os custos após a retirada de cada viagem.

3.4.3. Remoção Shaw

A terceira e última remoção construída foi a Remoção Shaw que utiliza o cálculo de similaridade. Nela, as viagens são comparadas e as mais semelhantes são removidas da solução. A similaridade é calculada por meio de comparações entre as viagens. Ela é calculada comparando a distância entre os pontos inicial e final das viagens, assim como os horários de início e fim. Quanto mais próximas são essas grandezas, menor é a pontuação dessas viagens e maior é a similaridade entre elas. O cálculo da similaridade entre as viagens é apresentado a seguir nas equações (2) a (5).

EQUAÇÃO DE SIMILARIDADE ENTRE DUAS VIAGENS	
$DH(i, j) = \alpha sim \times (TPi - TPj) + (TTi - TTj) $	(2)
$D(i, j) = \beta sim \times (TTi - TPi) + (TTj - TPj) $	(3)
$DP(i, j) = \gamma sim \times DLPij + DLTij $	(4)
$SIMILARIDADE(i, j) = DH(i, j) + D(i, j) + DP(i, j)$	(5)

Como pode ser visto acima, a equação da similaridade é composta por três parcelas (2) – (4), de forma que a soma das mesmas (5) se aproxima de zero para viagens com maior similaridade. A parcela $DH(i, j)$ representa a diferença entre os tempos de partida TP e tempo de término TT das viagens i e j . Essa primeira parcela é multiplicada por um parâmetro αsim . A segunda e terceira parcelas da equação são multiplicadas pelos coeficientes βsim e γsim respectivamente. A diferença entre a duração as viagens i e j é representada por $D(i, j)$. Já $DP(i, j)$ é a parcela que calcula a distância entre os terminais de início das viagens i e j , denotado por $DLPij$, e a distância entre os locais de término $DLTij$ das viagens i e j . A soma dessas três parcelas resulta na similaridade entre duas viagens. Essa função foi utilizada dentro da heurística de remoção, cujo pseudocódigo é apresentado na Figura 6.

Figura 6 – Pseudocódigo da Heurística de Remoção Shaw

REMOÇÃO SHAW ($q \in \mathbb{N}$, $s \in \text{soluções}$, $p_{\text{similaridade}}$)
<p>D = lista de viagens removidas por similaridade; L = lista de valores de similaridade entre as viagens;</p> <ol style="list-style-type: none"> 1. <i>retira-se aleatoriamente uma viagem r da solução;</i> 2. <i>insere r em D;</i> 3. Enquanto ($D < q$) { 4. r = <i>viagem selecionada aleatoriamente de D;</i> 5. L = <i>Cálculo_Similaridade (r, todas viagens da solução);</i> 6. <i>Ordena L (crescente, mais similares primeiros);</i> 7. y = <i>número aleatório entre $[0, 1)$;</i> 8. $D = DU\{L[y^p_{\text{similaridade}} L]\}$; 9. } Fim_Enquanto 10. <i>Retira da solução s as viagens contidas em D;</i>

Fonte: Adaptado de Ropke e Pisinger (2006)

As remoções foram implementadas para destruir a solução. A cada iteração da heurística, uma dada quantidade q de viagens é removida e posteriormente reinseridas na solução. No LNS é definido *a priori* a heurística de remoção e a heurística de inserção a ser utilizada ao longo da busca toda. Por outro lado, no ALNS, as heurísticas podem ser substituídas durante a busca, em função da eficiência de cada heurística. A seguir são apresentadas as heurísticas de inserção das viagens na solução que sofreu a remoção anteriormente.

3.5. Heurísticas de Inserção de Viagens

Para reconstruir a solução procurando minimizar o custo final, foram criadas duas heurísticas de inserção: a gulosa e a de arrependimento.

3.5.1. Inserção Gulosa

A inserção gulosa segue o princípio de construção da solução inicial. O algoritmo percorre o banco de viagens, retirando uma por vez, e procurando o melhor local para a inserção da mesma tendo em vista minimizar o custo total. Após percorrer todas as possibilidades, a viagem é inserida na melhor posição da solução e retirada do banco de viagens. O procedimento se repete até que o banco de viagens esteja vazio. O pseudocódigo dessa heurística de construção pode ser visto na Figura 7.

Figura 7 – Pseudocódigo da Heurística de Inserção Gulosa

INSERÇÃO GULOSA ($s \in \text{soluções}$, $vg \in \text{Banco}$)

```

1. Enquanto ( $|\text{Banco}| > 0$ ) {
2.   Para ( $vg = 1$ ;  $vg \leq |\text{Banco}|$ ;  $vg++$ ) {
3.     Se (inserção de  $vg$  em algum veículo da solução  $s$  for possível) {
4.       retira  $vg$  do Banco;
5.       Insira  $vg$  na solução  $s$  no veículo que produz a menor variação de custo;
6.        $|\text{Banco}| = |\text{Banco}| - 1$ 
7.     } Senão {
8.       criar novo veículo para  $s$ ;
9.       insira  $vg$  no novo veículo criado;
10.    } Fim_Se_Senão
11.  } Fim_Para
12. } Fim_Enquanto

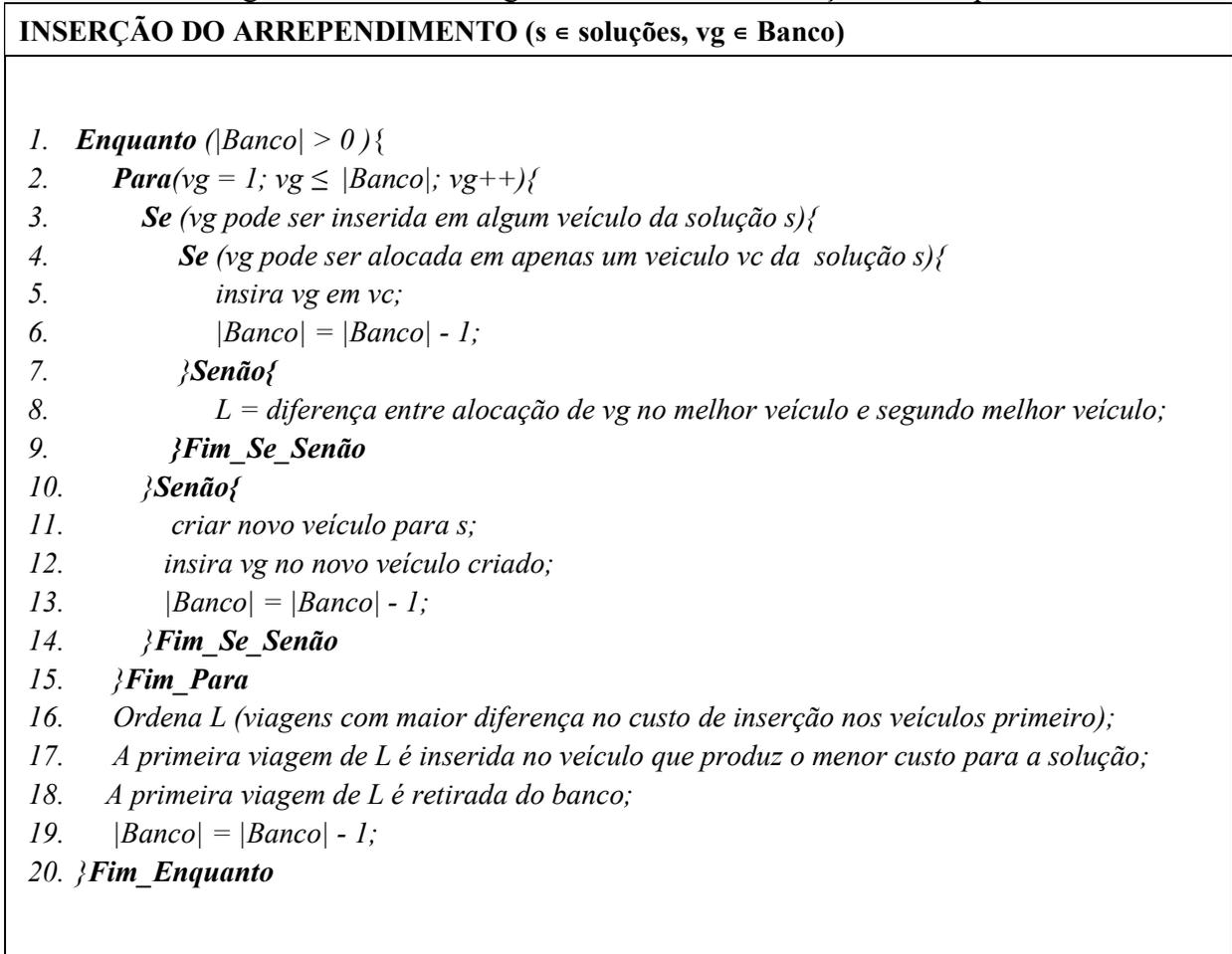
```

3.5.2. Inserção de Arrependimento

Na inserção do arrependimento, para cada viagem i o algoritmo armazena e compara os custos por inserir a viagem i na sua primeira melhor posição e na sua segunda melhor posição, tendo em vista o custo da solução. As viagens que apresentarem as maiores diferenças de custos entre essas posições são inseridas primeiro e na posição de menor custo. Isto é feito para se evitar que uma viagem deixe de ser inserida na sua melhor posição e tenha que ser alocada à sua segunda melhor posição, gerando um custo muito elevado se comparado à melhor posição para a viagem em questão. Na Figura 8, pode ser visto o pseudocódigo da inserção de arrependimento.

As heurísticas de remoção e inserção compõem o algoritmo do LNS, que busca a otimização da solução inicial encontrada.

Figura 8 – Pseudocódigo da Heurística de Inserção do Arrependimento



3.6. Algoritmo LNS para o PPV

Uma vez desenvolvidas as heurísticas de remoção e inserção, deve-se criar algum mecanismo para combinar a destruição e a reconstrução das soluções. Para isso, neste trabalho foi implementado o algoritmo LNS. Essa heurística conta com uma heurística de remoção e outra de inserção, definidas e fixadas *a priori*, para destruir a solução inicial e tentar reconstruí-la otimizando-a. Na Figura 9 pode ser visto o esquema de funcionamento do LNS, enquanto que o pseudocódigo dessa implementação pode ser observado na Figura 10.

Figura 9 – Esquema de funcionamento do LNS

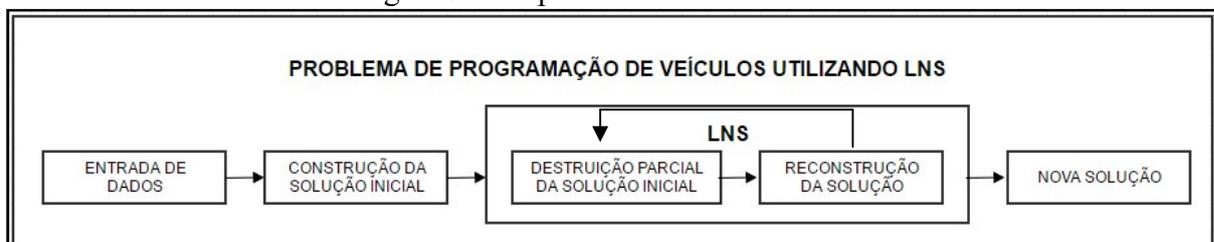


Figura 10 – Pseudocódigo da Metaheurística LNS

LNS ($q \in \mathbb{N}$, $s \in \text{soluções}$)

1. *Solução* $s_{\text{melhor}} = s$;
2. **Enquanto** (*tempo de execução* < *tempo máximo de execução*) {
3. $s' = s$;
4. *retira* q *viagens de* s' ;
5. *reinsere as viagens removidas em* s' ;
6. *custo_atual* = *custo da solução atual*;
7. **Se** (*custo de* $s' <$ *custo de* s_{melhor}) {
8. $s_{\text{melhor}} = s'$;
9. **}Fim_Se**
10. **}Fim_Enquanto**

Fonte: Adaptado de Ropke e Pisinger (2006)

Ao fim da implementação do LNS, as várias heurísticas construídas foram combinadas, de forma que foram testadas várias versões do LNS, cada uma com uma combinação distinta de heurísticas de remoção e inserção. Na seção adiante serão detalhados os procedimentos para execução de testes de forma mais clara.

4. APRESENTAÇÃO E DISCUSSÃO DOS RESULTADOS

Ao término da implementação da heurística LNS, foram realizados testes de desempenho das mesmas, resolvendo problemas reais de uma série de empresas de transporte público da cidade de Belo Horizonte, MG. As Tabelas 2, 3 e 4 apresentam os resultados obtidos nestes testes computacionais para os problemas de segunda-feira, sábado e domingo respectivamente. As heurísticas implementadas são representadas pelos seguintes números na tabela: 1) Remoção Aleatória, 2) Remoção da Viagem de Pior Custo - versão 1, 3) Remoção da Viagem de Pior Custo - versão 2, 4) Remoção Shaw, 5) Inserção Gulosa, e 6) Inserção do Arrependimento. Os testes foram realizados em nove problemas distintos, sendo três garagens diferentes para cada dia, segunda-feira, sábado e domingo. No total foram realizados 10 testes de 30 minutos para cada par de heurística em cada problema.

A solução ótima apresentada nas Tabelas 2, 3 e 4 é uma solução exata, encontrada aplicando o método *ArcGen* de Silva (2001), que utiliza algoritmo de Fluxo em Redes para a sua obtenção. O que se busca com o LNS aqui apresentado é alcançar a mesma solução ou uma solução mais próxima possível da solução ótima conhecida. A coluna N° de Melhoras apresentada na Tabela 2 mostra em quantos testes foram encontradas melhores soluções em relação a FO para cada par de heurísticas. O valor na coluna Melhor FO representa qual o menor custo encontrado para o problema considerando todos os testes executados para todas as combinações do LNS. O LNS foi calibrado com os seguintes valores para os parâmetros apresentados nas seções anteriores: $q = 20\%$ do total de viagens; $p_{pior} = 6$; $p_{similaridade} = 6$; $\alpha_{sim} = 1$; $\beta_{sim} = 1$; $\gamma_{sim} = 1$. Por falta de tempo para realizar um estudo detalhado que permitisse o refinamento dos parâmetros, os mesmos foram obtidos da literatura, através do estudo de Ropke e Pisinger (2007).

A linguagem C/C++ foi adotada para a criação do programa e os testes foram executados num computador com um processador Intel Core i7 3370 CPU @ 3.40GHz, memória RAM de 12,0 GB e sistema operacional Windows 10 Pro de 64 Bits. Os problemas testados foram para os seguintes dias da semana: segunda-feira (SEG), sábado (SAB) e domingo (DOM) e para cada dia da semana, os veículos iniciam a operação a partir das garagens G02, G27, G48 e G61.

Para o problema SEG-G27 foram necessários 13 veículos para realizar todas as viagens da TH, tanto para o método exato, quanto para a solução inicial deste trabalho e o LNS. A diferença se apresentou neste caso em relação ao valor da FO: 17.818 para a solução inicial, 17.705 para o LNS e 15.012 para a solução ótima. No problema SEG-G48 a solução inicial utilizou 38 veículos, contando com um custo de 51.620. A melhor solução encontrada pelo LNS foi 14,2% mais cara do que a solução ótima, alcançando um valor de 45.911 e 33 veículos contra 40.212 e 32 veículos. Na solução inicial implementada neste trabalho foi preciso alocar 38 veículos para realizar todas as viagens. O custo desta solução ficou em 51.620. Com o LNS foram necessários 33 veículos, com o custo em 45911, 14,2 % mais caro do que a solução ótima. Já no problema SEG-G61, com a solução exata foram necessários 72 veículos para realizar todas as viagens, gerando um custo total de 88.341. A solução inicial desenvolvida neste trabalho utilizou 77 veículos e a FO, neste caso, foi de 103.399. Com o LNS foi possível reduzir o número de veículos da solução inicial, chegando a 72, totalizando um custo de 101.116, o que representa um valor 14,5% maior do que a solução ótima.

Tabela 2 – Resultado dos testes do LNS para os problemas de segunda-feira

Problema	SOLUÇÃO ÓTIMA		SOLUÇÃO INICIAL		LNS			
	Melhor FO	Nº de Veículos	Custo FO	Nº de Veículos	Heurísticas	Nº de Melhoras	Melhor FO	Nº de Veículos
SEG G27 (98 viagens)	15012	13	17818	13	1 e 5	1	17705	13
					2 e 5	10		
					3 e 5	9		
					4 e 5	6		
					1 e 6	1		
					2 e 6	0		
					3 e 6	1		
4 e 6	9							
SEG G48 (468 viagens)	40212	32	51620	38	1 e 5	1	45911	33
					2 e 5	10		
					3 e 5	8		
					4 e 5	0		
					1 e 6	4		
					2 e 6	10		
					3 e 6	8		
4 e 6	2							
SEG G61 (1.038 viagens)	88341	72	103399	77	1 e 5	0	101116	74
					2 e 5	4		
					3 e 5	4		
					4 e 5	0		
					1 e 6	0		
					2 e 6	4		
					3 e 6	2		
4 e 6	0							

Para o problema SAB-G02 a solução inicial custou 27.051. Foram utilizados 23 veículos para construir esta solução. Com o LNS o número de veículos permaneceu igual, mas o custo caiu para 26.979 contra 26.536 da solução ótima, que também utilizou o mesmo número de veículos. Neste caso, o LNS foi 1,7% mais caro do que o método exato. Já no problema SAB-G27 na solução ótima foram necessários 7 veículos, que foi o mesmo número de veículos utilizados tanto na solução inicial deste trabalho quanto no LNS. A diferença neste caso foi o custo: 8.406 para a solução ótima, 9.373 na solução inicial e 9.114 no LNS. Para este problema a metaheurística foi 11,5% mais cara do que a solução ótima. No problema SAB-G61, na solução inicial a FO foi de 66.384 e foram utilizados 52 veículos. No LNS foram utilizados 48 veículos e a melhor FO ficou em 62.034, 12,2% mais caro que a solução inicial com custo de 55.285 e 43 veículos.

Tabela 3 – Resultado dos testes do LNS para os problemas de sábado

Problema	SOLUÇÃO ÓTIMA		SOLUÇÃO INICIAL		LNS			
	Melhor FO	Nº de Veículos	Custo FO	Nº de Veículos	Heurísticas	Nº de Melhoras	Melhor FO	Nº de Veículos
SAB G02 (172 viagens)	26536	23	27051	23	1 e 5	0	26979	23
					2 e 5	1		
					3 e 5	0		
					4 e 5	0		
					1 e 6	0		
					2 e 6	5		
					3 e 6	1		
4 e 6	3							
SAB G27 (69 viagens)	8406	7	9373	7	1 e 5	0	9114	7
					2 e 5	10		
					3 e 5	6		
					4 e 5	2		
					1 e 6	7		
					2 e 6	10		
					3 e 6	10		
4 e 6	7							
SAB G61 (769 viagens)	55285	43	66384	52	1 e 5	0	62076	48
					2 e 5	7		
					3 e 5	5		
					4 e 5	0		
					1 e 6	0		
					2 e 6	10		
					3 e 6	10		
4 e 6	0							

Para o problema DOM-G02 foram necessários 11 veículos para realizar todas as viagens, tanto na solução inicial, quanto no LNS e na solução ótima. O custo porém foi de 14.005, 13.517 e 13.340 respectivamente. O LNS ficou 1,3% mais caro que a solução ótima neste

caso. No problema DOM-G27 a solução ótima utiliza 5 veículos, totalizando um custo de 6.143. A solução inicial encontrada neste trabalho também utilizou 5 veículos, encontrando uma FO de 6.816. Com o LNS, o custo da solução diminuiu para 6.796 para o mesmo número de veículos, o que representa um custo 10,6% superior ao da solução ótima. Por fim, no problema DOM-G61 foram utilizados 31 veículos na solução inicial, totalizando um custo de 39.820. No LNS foram necessários 26 veículos e o custo foi de 35.132. Foram utilizados 24 veículos na solução ótima e a FO totalizou 30.963. Neste problema o LNS foi 13,5% mais caro que a solução ótima.

Tabela 4 – Resultado dos testes do LNS para os problemas de domingo

Problema	SOLUÇÃO ÓTIMA		SOLUÇÃO INICIAL		LNS			
	Melhor FO	Nº de Veículos	Custo FO	Nº de Veículos	Heurísticas	Nº de Melhoras	Melhor FO	Nº de Veículos
DOM G02 (90 viagens)	13340	11	14005	11	1 e 5	0	13517	11
					2 e 5	10		
					3 e 5	10		
					4 e 5	0		
					1 e 6	1		
					2 e 6	10		
					3 e 6	10		
					4 e 6	10		
DOM G27 (52 viagens)	6143	5	6816	5	1 e 5	8	6796	5
					2 e 5	10		
					3 e 5	10		
					4 e 5	5		
					1 e 6	4		
					2 e 6	2		
					3 e 6	2		
					4 e 6	3		
DOM G61 (538 viagens)	30963	24	39820	31	1 e 5	0	34428	26
					2 e 5	10		
					3 e 5	8		
					4 e 5	0		
					1 e 6	0		
					2 e 6	10		
					3 e 6	10		
					4 e 6	0		

Nas Tabelas de 5 a 13 estão os resultados de todos os testes realizados para cada par de heurísticas em cada problema. Nelas podem ser vistos, em negrito, os melhores valores da FO encontrada em cada teste. Essas tabelas foram utilizadas para gerar os gráficos nas Figuras 11 a 19. Nos gráficos podem ser visualizados os desempenhos para cada versão do LNS em cada problema.

Tabela 5 – Resultado dos testes do LNS para o problema SEG-G27

SEGUNDA-FEIRA (G27)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	17.818	17.706	17.758	17.818	17.818	17.818	17.818	17.798
2	17.818	17.715	17.744	17.818	17.818	17.818	17.818	17.751
3	17.818	17.713	17.803	17.797	17.818	17.818	17.818	17.754
4	17.818	17.721	17.754	17.776	17.818	17.818	17.818	17.739
5	17.818	17.720	17.740	17.818	17.818	17.818	17.818	17.818
6	17.818	17.717	17.801	17.738	17.818	17.818	17.818	17.749
7	17.818	17.706	17.800	17.808	17.806	17.818	17.775	17.757
8	17.818	17.705	17.818	17.818	17.818	17.818	17.818	17.730
9	17.770	17.717	17.807	17.745	17.818	17.818	17.818	17.768
10	17.818	17.720	17.758	17.774	17.818	17.818	17.818	17.738
Nº de Viagens	98	Custo da Solução Inicial			17.818			

Tabela 6 – Resultado dos testes do LNS para o problema SEG-G48

SEGUNDA-FEIRA (G48)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	51.620	48.175	48.233	51.620	51.620	48.504	47.225	51.026
2	51.620	48.577	50.130	51.620	49.464	45.911	47.584	51.620
3	51.620	49.096	49.212	51.620	51.620	49.572	49.549	51.620
4	51.620	47.524	49.878	51.620	51.620	49.754	48.814	51.620
5	51.620	49.815	51.620	51.620	51.620	47.770	51.620	51.620
6	51.620	46.178	51.620	51.620	50.986	47.753	46.202	51.620
7	51.620	49.174	49.089	51.620	51.099	46.738	51.620	51.620
8	51.620	48.324	49.558	51.620	49.327	49.774	47.310	50.340
9	49.712	48.798	50.249	51.620	51.620	49.446	47.938	51.620
10	51.620	48.142	49.664	51.620	51.620	47.385	48.102	51.620
Nº de Viagens	468	Custo da Solução Inicial			51.620			

Tabela 7 – Resultado dos testes do LNS para o problema SEG-G61 (continua)

SEGUNDA-FEIRA (G61)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	103.399	102.943	103.202	103.399	103.399	103.399	103.285	103.399
2	103.399	101.961	103.399	103.399	103.399	103.258	103.258	103.399
3	103.399	103.399	103.399	103.399	103.399	102.660	103.399	103.399
4	103.399	103.450	103.399	103.399	103.399	103.399	103.399	103.399
5	103.399	103.399	102.422	103.399	103.399	103.399	103.399	103.399
6	103.399	103.399	103.399	103.399	103.399	103.399	103.399	103.399
7	103.399	103.399	103.399	103.399	103.399	103.285	103.399	103.399

Tabela 7 – Resultado dos testes do LNS para o problema SEG-G61 (conclusão)

8	103.399	103.399	102.406	103.399	103.399	103.399	103.399	103.399
9	103.399	103.399	103.399	103.399	103.399	103.399	103.399	103.399
10	103.399	101.116	103.399	103.399	103.399	102.190	103.399	103.399
Nº de Viagens	1.038	Custo da Solução Inicial			103.399			

Tabela 8 – Resultado dos testes do LNS para o problema SAB-G02

SÁBADO (G02)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	27.051	27.051	27.051	27.051	27.051	27.051	27.051	27.051
2	27.051	27.051	27.051	27.051	27.051	26.994	27.051	26.979
3	27.051	27.051	27.051	27.051	27.051	27.051	27.051	27.051
4	27.051	27.051	27.051	27.051	27.051	27.051	27.046	27.043
5	27.051	27.024	27.051	27.051	27.051	26.998	27.051	27.051
6	27.051	27.051	27.051	27.051	27.051	27.051	27.051	27.020
7	27.051	27.051	27.051	27.051	27.051	26.993	27.051	27.051
8	27.051	27.051	27.051	27.051	27.051	27.051	27.051	27.051
9	27.051	27.051	27.051	27.051	27.051	27.029	27.051	27.051
10	27.019	27.051	27.051	27.051	27.051	27.027	27.051	27.051
Nº de Viagens	172	Custo da Solução Inicial			27.051			

Tabela 9 – Resultado dos testes do LNS para o problema SAB-G27

SÁBADO (G27)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	9.373	9.150	9.132	9.373	9.151	9.131	9.131	9.131
2	9.373	9.143	9.373	9.373	9.131	9.131	9.131	9.151
3	9.373	9.155	9.221	9.373	9.151	9.114	9.131	9.373
4	9.373	9.162	9.186	9.300	9.131	9.131	9.131	9.259
5	9.373	9.151	9.373	9.373	9.131	9.131	9.131	9.296
6	9.373	9.144	9.180	9.373	9.373	9.131	9.131	9.373
7	9.373	9.161	9.373	9.373	9.155	9.131	9.131	9.373
8	9.373	9.174	9.143	9.373	9.373	9.131	9.131	9.295
9	9.373	9.131	9.373	9.252	9.373	9.131	9.131	9.152
10	9.373	9.161	9.131	9.373	9.151	9.131	9.131	9.186
Nº de Viagens	69	Custo da Solução Inicial			9.373			

Tabela 10 – Resultado dos testes do LNS para o problema SAB-G61

SÁBADO (G61)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	66.384	64.471	66.384	66.384	66.384	62.193	63.766	66.384
2	66.384	65.892	66.384	66.384	66.384	63.512	64.875	66.384
3	66.384	66.384	66.384	66.384	66.384	62.455	63.837	66.384
4	66.384	66.384	66.384	66.384	66.384	62.183	62.423	66.384
5	66.384	63.985	65.260	66.384	66.384	62.337	63.321	66.384
6	66.384	63.664	65.126	66.384	66.384	62.871	63.972	66.384
7	66.384	64.819	66.384	66.384	66.384	62.287	62.601	66.384
8	66.384	64.593	64.776	66.384	66.384	62.506	64.700	66.384
9	66.384	64.937	65.107	66.384	66.384	64.034	64.596	66.384
10	66.384	66.384	64.997	66.384	66.384	62.859	62.076	66.384
Nº de Viagens	769	Custo da Solução Inicial			66.384			

Tabela 11 – Resultado dos testes do LNS para o problema DOM-G02

DOMINGO (G02)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	14.005	13.721	13.634	14.005	14.005	13.610	13.665	13.956
2	14.005	13.696	13.821	14.005	14.005	13.517	13.699	13.720
3	14.005	13.890	13.728	14.005	14.005	13.702	13.657	13.971
4	14.005	13.797	13.917	14.005	14.005	13.560	13.519	13.768
5	14.005	13.580	13.828	14.005	14.005	13.637	13.688	13.969
6	14.005	13.712	13.734	14.005	14.005	13.770	13.673	13.811
7	14.005	13.745	13.762	14.005	13.966	13.517	13.598	13.910
8	14.005	13.746	13.754	14.005	14.005	13.617	13.616	13.896
9	14.005	13.722	13.828	14.005	14.005	13.562	13.560	13.811
10	14.005	13.729	13.858	14.005	14.005	13.607	13.646	13.780
Nº de Viagens	90	Custo da Solução Inicial			14.005			

Tabela 12 – Resultado dos testes do LNS para o problema DOM-G27 (continua)

DOMINGO (G27)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	6.796	6.796	6.796	6.796	6.816	6.796	6.796	6.816
2	6.816	6.796	6.796	6.816	6.796	6.816	6.816	6.816
3	6.796	6.796	6.796	6.796	6.816	6.816	6.816	6.801
4	6.801	6.796	6.796	6.816	6.796	6.816	6.816	6.796
5	6.796	6.796	6.796	6.816	6.816	6.816	6.816	6.801
6	6.801	6.796	6.796	6.796	6.796	6.816	6.816	6.816
7	6.801	6.796	6.796	6.796	6.816	6.816	6.816	6.816

Tabela 12 – Resultado dos testes do LNS para o problema DOM-G27 (conclusão)

8	6.801	6.796	6.796	6.796	6.816	6.816	6.816	6.796
9	6.796	6.796	6.796	6.816	6.816	6.816	6.796	6.816
10	6.816	6.796	6.796	6.816	6.796	6.796	6.816	6.801
Nº de Viagens	52	Custo da Solução Inicial			6.816			

Tabela 13 – Resultado dos testes do LNS para o problema DOM-G61

DOMINGO (G61)								
Custo da Solução								
Teste/Heurísticas	1 e 5	2 e 5	3 e 5	4 e 5	1 e 6	2 e 6	3 e 6	4 e 6
1	39.820	36.569	39.727	39.820	39.820	34.641	34.428	39.820
2	39.820	36.487	37.975	39.820	39.820	34.712	35.957	39.820
3	39.820	37.126	36.845	39.820	39.820	35.892	34.869	39.820
4	39.820	36.983	39.820	39.820	39.820	36.460	36.494	39.820
5	39.820	38.581	39.820	39.820	39.820	36.466	36.111	39.820
6	39.820	37.503	36.960	39.820	39.820	36.770	35.928	39.820
7	39.820	36.565	36.568	39.820	39.820	37.323	36.352	39.820
8	39.820	36.465	36.612	39.820	39.820	35.132	37.171	39.820
9	39.820	36.423	36.805	39.820	39.820	36.309	36.784	39.820
10	39.820	37.057	37.278	39.820	39.820	36.548	36.496	39.820
Nº de Viagens	538	Custo da Solução Inicial			39.820			

Figura 11 – Desempenho das Heurísticas no Problema SEG-G27

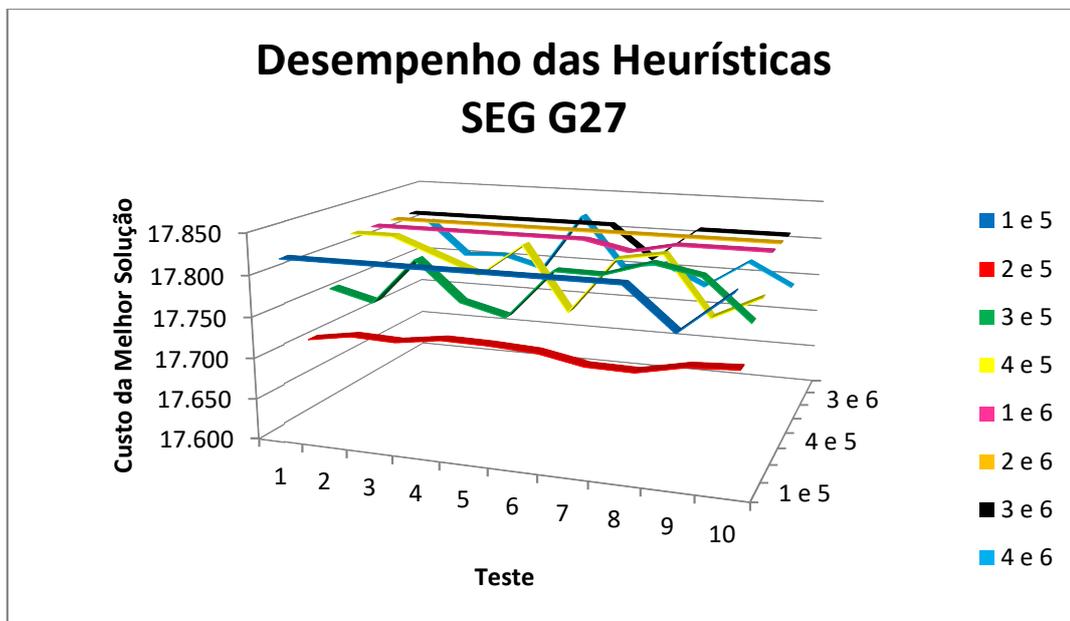


Figura 12 – Desempenho das Heurísticas no Problema SEG-G48

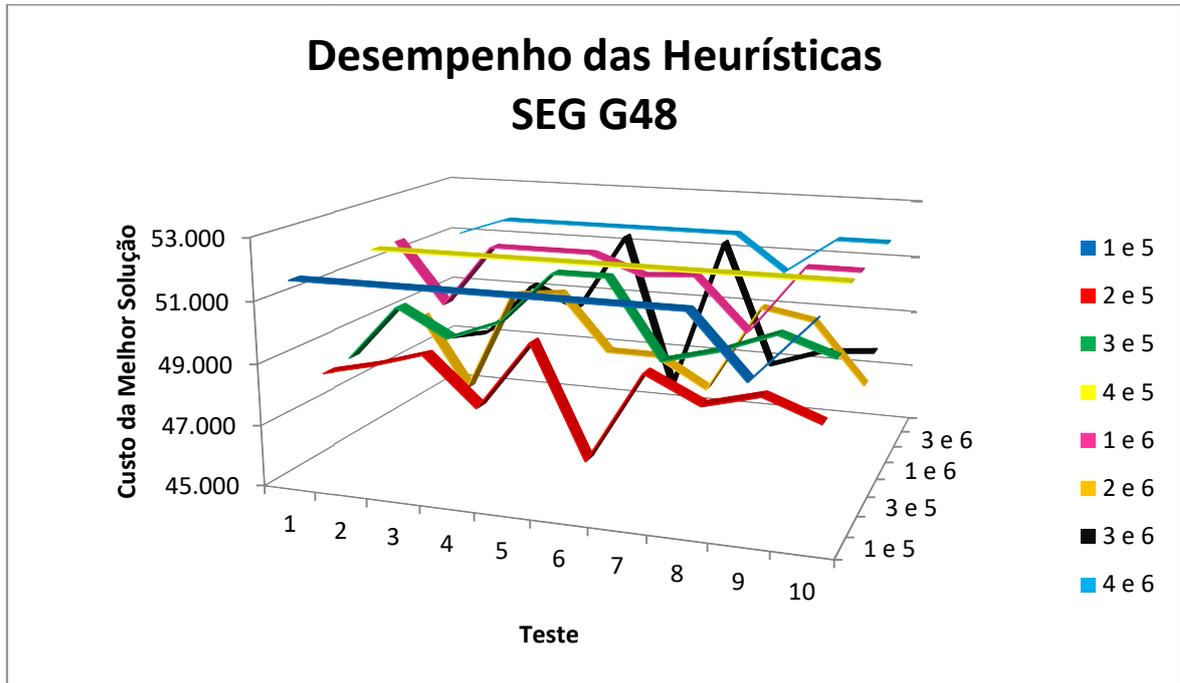


Figura 13 – Desempenho das Heurísticas no Problema SEG-G61

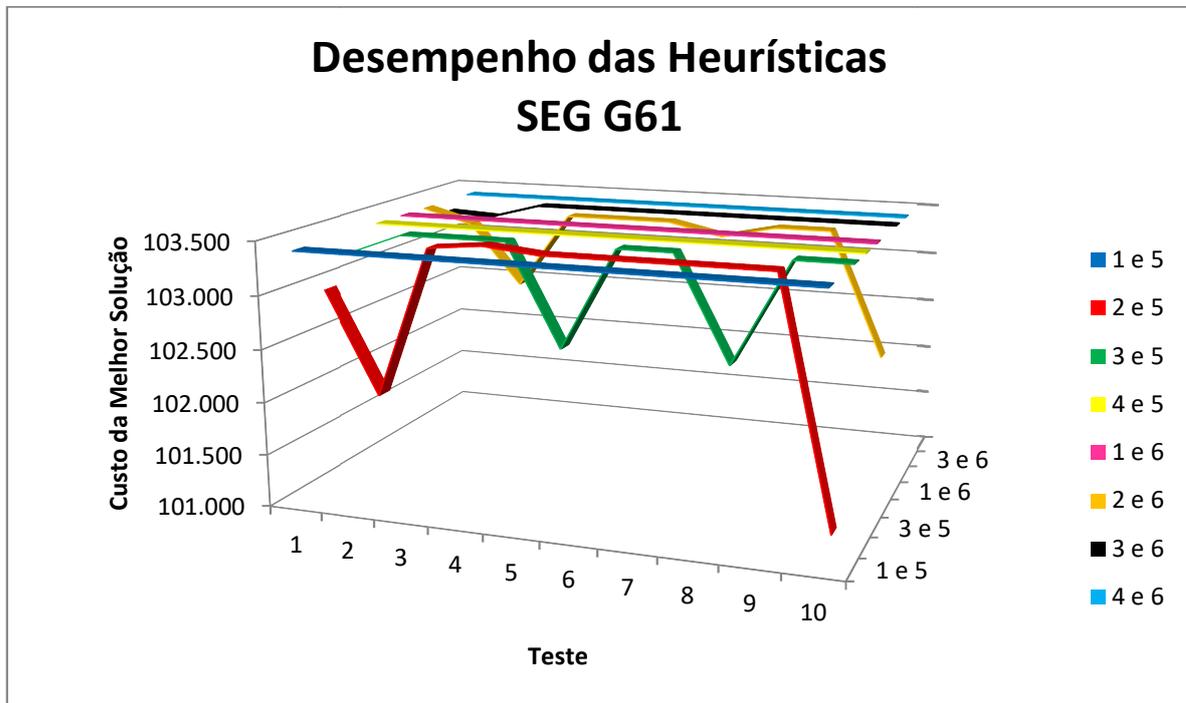


Figura 14 – Desempenho das Heurísticas no Problema SAB-G02

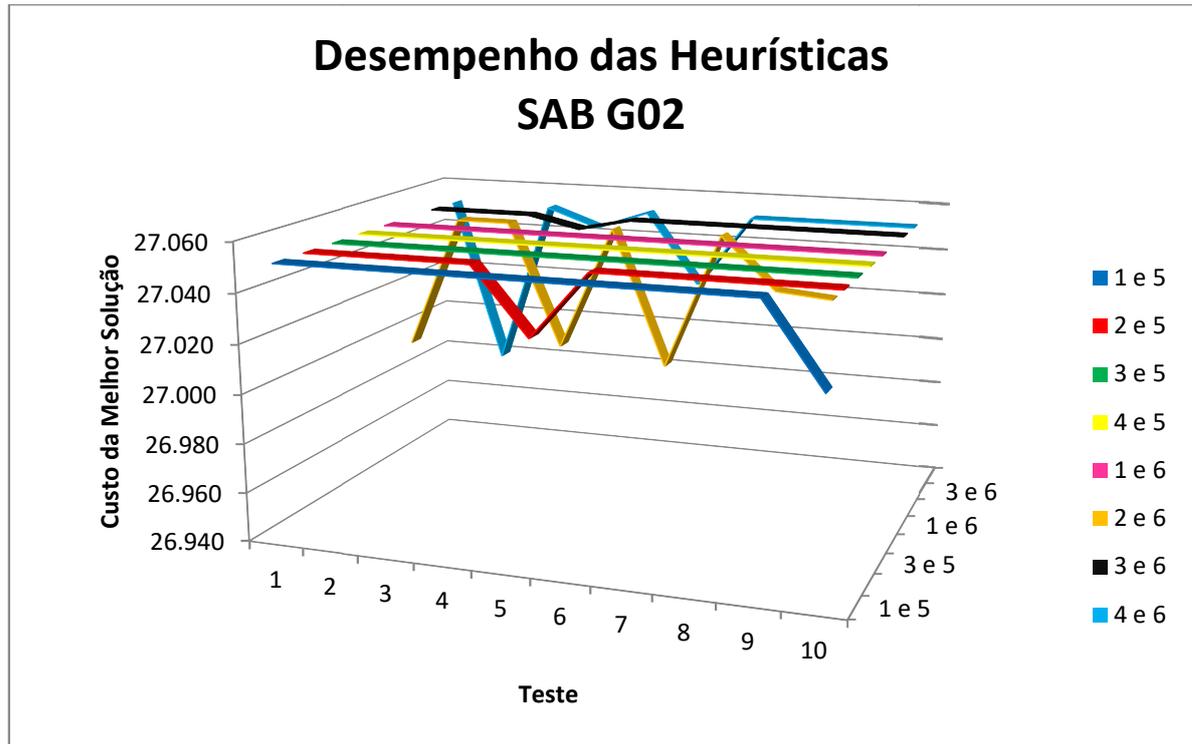


Figura 15 – Desempenho das Heurísticas no Problema SAB-G27

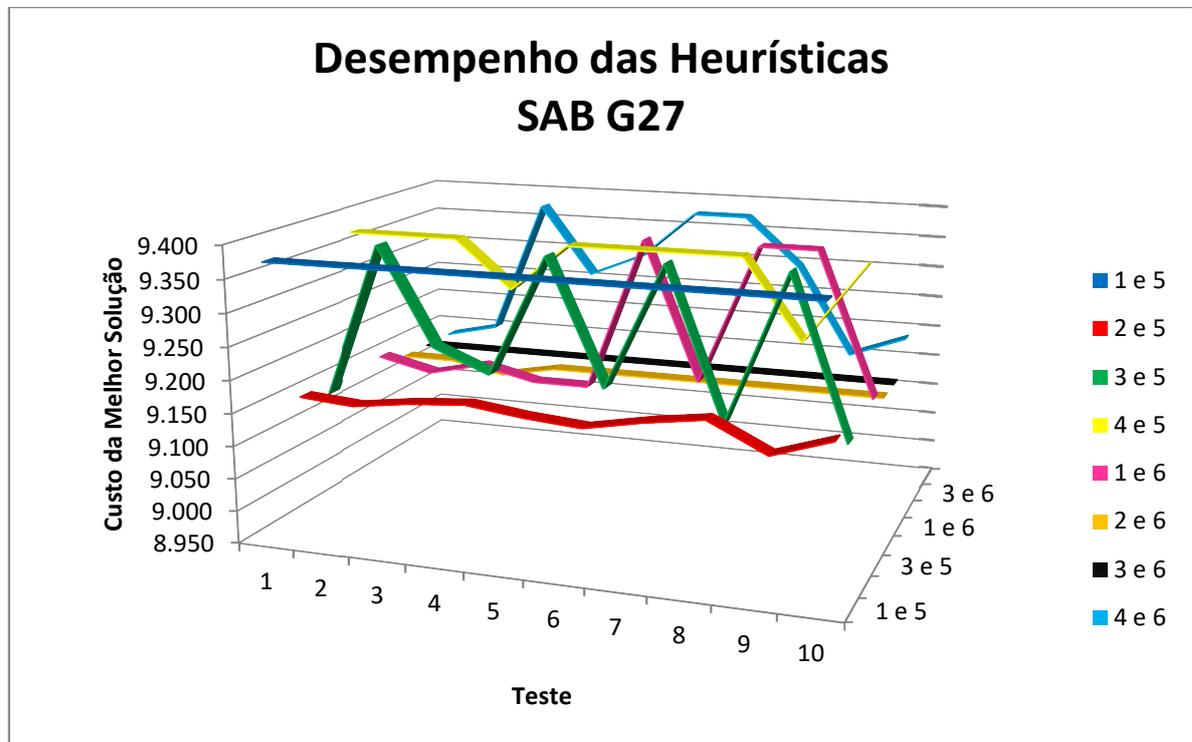


Figura 16 – Desempenho das Heurísticas no Problema SAB-G61

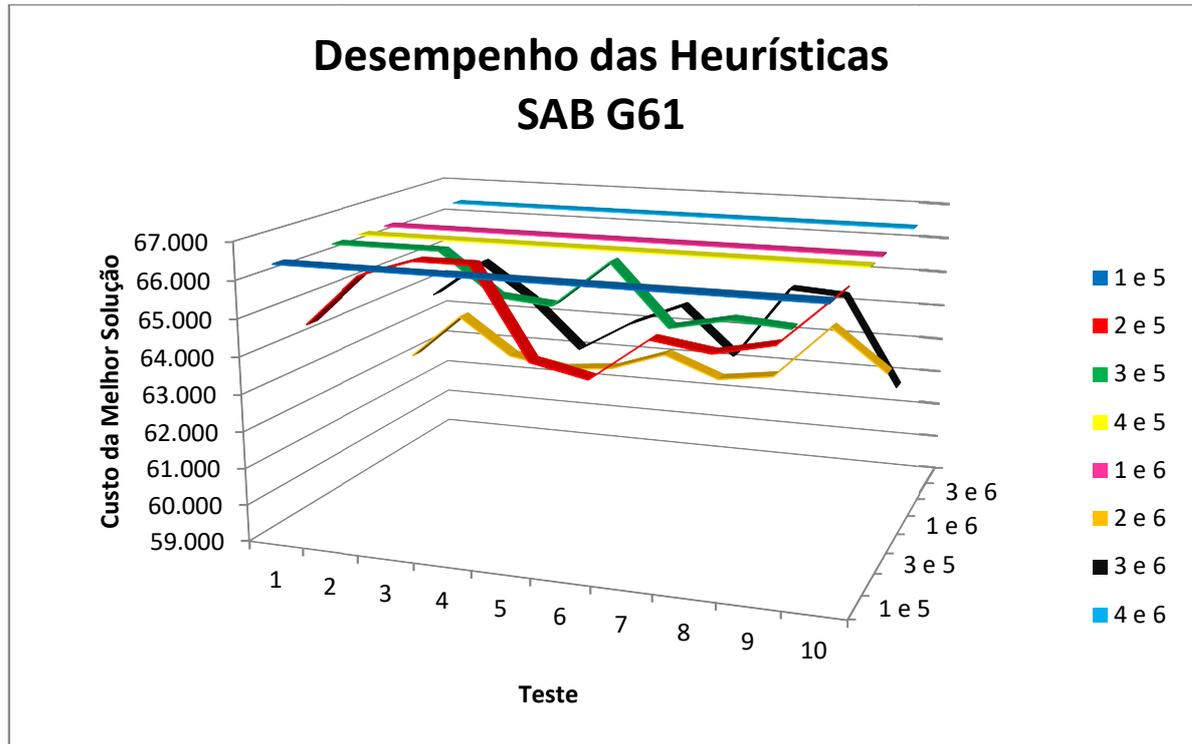


Figura 17 – Desempenho das Heurísticas no Problema DOM-G02

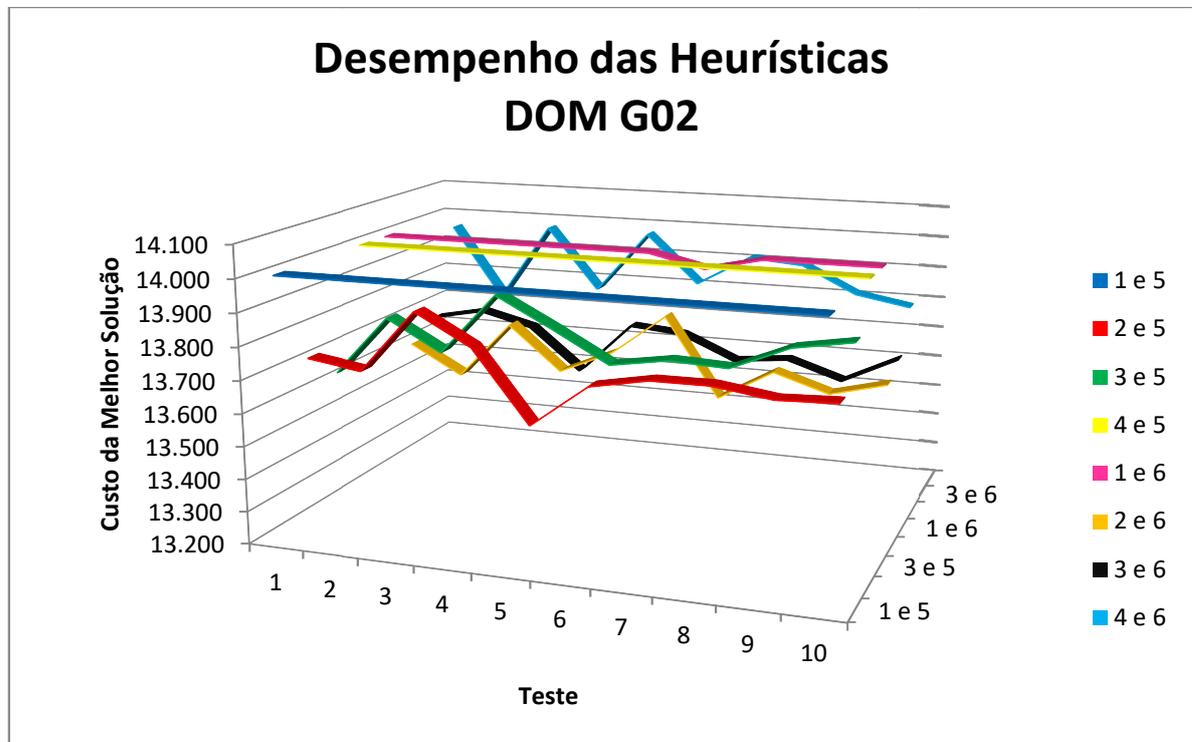


Figura 18 – Desempenho das Heurísticas no Problema DOM-G27

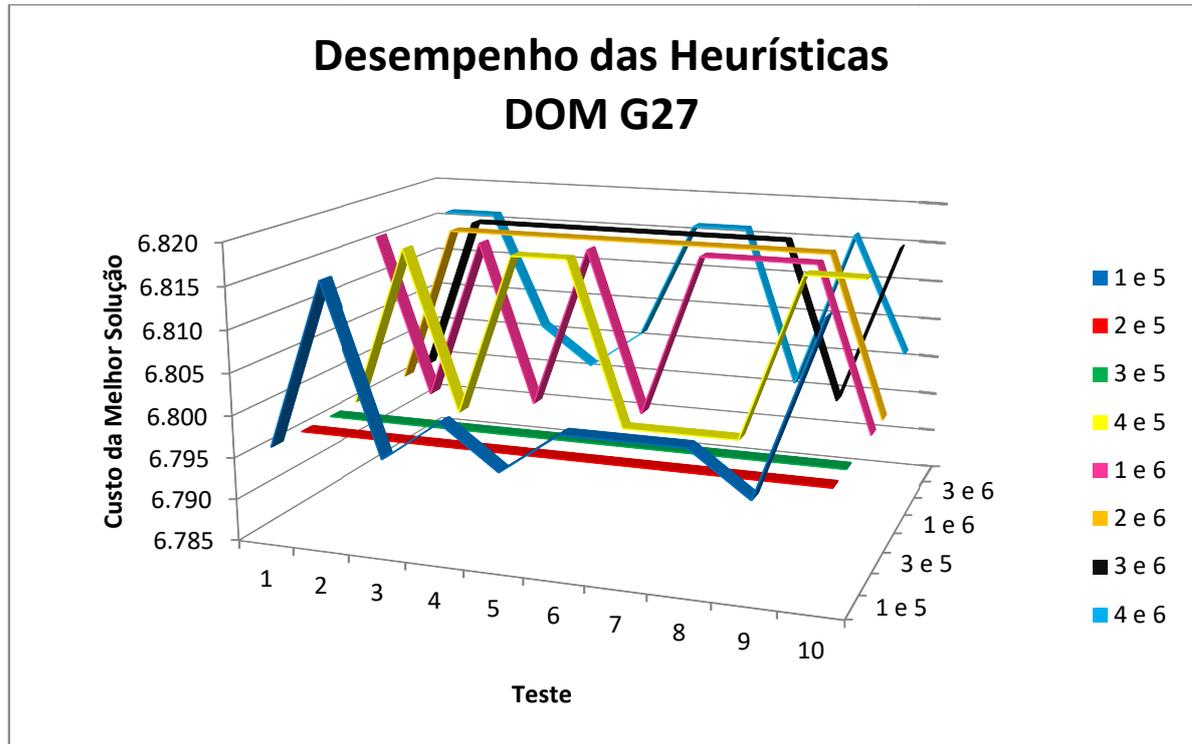
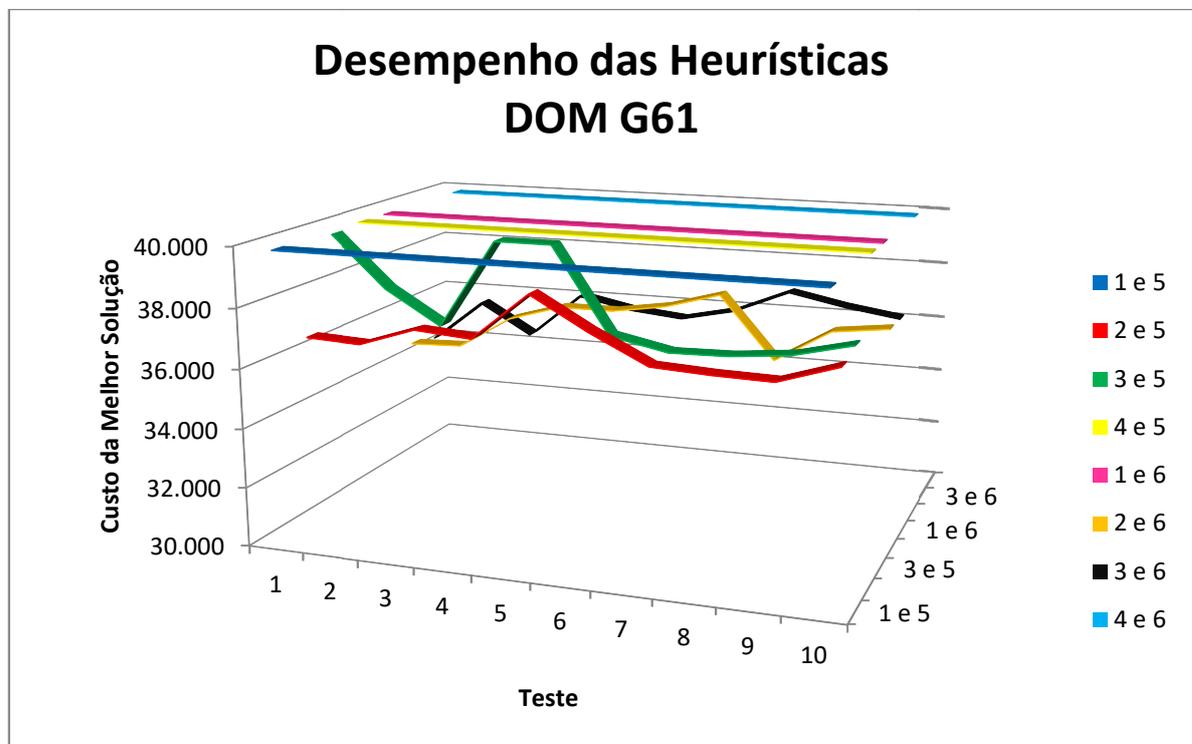


Figura 19 – Desempenho das Heurísticas no Problema DOM-G61



Analisando os resultados contidos nas Tabela 2 até a Tabela 13 e dos gráficos da Figura 11 até Figura 19 podemos perceber que as melhores versões do LNS foram com as combinações 2-

5) Remoção do Pior (versão 1) - Inserção Gulosa e 2-6) Remoção do Pior (versão 1) - Inserção de Arrependimento. Na combinação 2-5) a maior parte dos testes obteve melhora, com 72 testes retornando soluções melhores do que a inicial. Porém na combinação 2-6) foram encontrados os melhores valores da FO. Em quatro dos nove testes as melhores soluções foram encontradas por essa combinação. A Tabela 14 compara o número de veículos utilizados na solução adotada pelas empresas de transporte com o LNS e o *ArcGen*.

Tabela 14 – Número de veículos utilizados para cada solução

	SEG G27	SEG G48	SEG G61	SAB G02	SAB G27	SAB G61	DOM G02	DOM G27	DOM G61
Solução das Empresas	14	34	74	23	9	60	11	6	39
LNS	13	33	74	23	7	48	11	5	26
Solução Ótima	13	32	72	23	7	43	11	5	24
% LNS x Ótimo	0	3,12	2,78	0	0	11,62	0	0	8,33

Os veículos representam o maior custo para a FO, portanto é de suma importância buscar a redução da frota para realização das viagens. Além disso, diminuir o número de veículos tem um impacto ambiental positivo, uma vez que a emissão de gases poluentes é reduzida. Com o método proposto neste trabalho, em 6 de 9 ocasiões foi possível reduzir o número de veículos em relação a solução adotada pelas empresas. Além disso, em 5 dos 9 problemas, o número de veículos encontrado pelo LNS foi o mesmo que o utilizado pela solução ótima.

5. CONCLUSÃO

Em todos os testes executados, independente do problema que foi abordado e qual par de heurísticas que compuseram o LNS, houve melhora em relação à solução inicial encontrada. No pior caso, a melhor solução encontrada pelo LNS foi aproximadamente 0,3% melhor do que a solução inicial. Isso ocorreu para o problema SAB-G02. O melhor percentual de melhora ocorreu para o problema DOM-G61. Neste caso a melhor solução encontrada pelo LNS foi aproximadamente 12% melhor do que a solução inicial encontrada para o problema.

Como foi citado na seção anterior, a heurística de Remoção da Viagem de Pior Custo (versão 1) foi a mais eficiente segundo os testes realizados. Esta remoção combinada com qualquer uma das inserções teve um percentual elevado de melhoras nas soluções. Das 180 vezes em que foi utilizada, em 133 a solução inicial foi melhorada, otimizando a FO em 73,9% dos testes. A Remoção da Viagem de Pior Custo (versão 1) combinada com a Inserção Gulosa encontrou soluções melhores mais vezes do que combinada com a Inserção do Arrependimento. Entretanto, nessa última combinação foram encontrados os melhores valores de FO para cada problema.

Analisando as inserções separadamente percebemos que a Inserção de Arrependimento pode ser considerada melhor do que a Inserção Gulosa, pois encontrou melhores soluções 166 vezes contra 155, independente de qual remoção foi utilizada. Portanto, podemos concluir que a melhor versão do LNS testada foi a 2-6) com a Remoção da Viagem de Pior Custo (versão 1) combinada com a Inserção do Arrependimento.

Apesar de otimizar a solução em todos os testes, acredita-se que o desempenho do LNS implementado neste trabalho ainda pode ser melhorado, conseguindo se aproximar mais das soluções ótimas produzidas pelo método exato. No pior caso, o custo da solução pelo LNS foi 14,5% maior que a solução ótima. No melhor caso foi 1,3% mais cara. Porém, foram apenas duas vezes que a solução proposta neste trabalho ficou abaixo dos 2% mais cara do que o *ArcGen*.

Dessa maneira, acredita-se que para trabalhos futuros, pode ser realizada a calibragem do LNS, para se encontrar o melhor valor para os parâmetros utilizados no código e adicionar

perturbações no algoritmo, para diversificar a busca no espaço de soluções. Outro método que pode ser adicionado neste trabalho para melhorar o desempenho da aplicação é a heurística de remoção média, apresentada por Martins (2017), onde foi a melhor heurística de destruição de soluções. Além disso, as heurísticas implementadas neste trabalho podem ser utilizadas para elaborar o método ALNS, que pode contribuir para uma melhora significativa no desempenho do sistema implementado.

6. REFERÊNCIAS BIBLIOGRÁFICAS

BARNHART, C.; JOHNSON, E. L.; NEMHAUSER, G. L.; SAVELSBERGH, M.W. P.; VANCE, P. H. Branch-and-price: column generation for solving huge integer programs. **Operations Research**, v. 46, p. 316–329, 1998.

CEDER, A.; WILSON, N. H. Bus network design. **Transportation Research Part B: Methodological**, v. 20, n. 4, p. 331-344, 1986.

IBARRA-ROJAS, O. J.; GIESEN, R.; RIOS-SOLIS, Y. A. An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. **Transportation Research Part B: Methodological**, v. 70, p. 35-46, 2014.

KLIEWER, N.; AMBERG, B.; AMBERG, B. Multiple depot vehicle and crew scheduling with time windows for scheduled trips. **Public Transport**, v. 3, n. 3, p. 213-244, 2012.

KWAN, R. K.; RAHIN, M. A. Object oriented bus scheduling – the BOOST system. **Computer Aided Scheduling of Public Transport**, p. 179-191, 1999.

MARTINS, L. C. **O uso da heurística Adaptive Large Neighborhood Search para resolver o Problema de Programação de Tripulações do Transporte Público**. 2017. 101 f. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Ouro Preto, Ouro Preto, 2017.

MING, W. E. I.; BO, S. U. N.; WENZHOU, J. I. N. Model and algorithm of regional bus scheduling with grey travel time. **Journal of Transportation Systems Engineering and Information Technology**, v. 12, n. 6, p. 106-112, 2012.

PATTNAIK, S. B.; MOHAN, S.; TOM, V. M. Urban bus transit route network design using genetic algorithm. **Journal of transportation engineering**, v. 124, n. 4, p. 368-375, 1998.

PEPIN, A. S.; DESAULNIERS, G.; HERTZ, A.; HUISMAN, D. A comparison of five heuristics for the multiple depot vehicle scheduling problem. **Journal of Scheduling**, v. 12, n. 1, p. 17-30, 2009.

ROPKE, S.; PISINGER, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. **Transportation Science**, v 40, n 4, p. 455-472, 2006.

SCHRIMPF, G.; SCHNEIDER, J.; STAMM-WILBRANDT, H.; DUECK, G. Record breaking optimization results using the ruin and recreate principle. **Journal of Computational Physics**, v. 159, n. 2, p. 139-171, 2000.

SILVA, G. P. **Uma metodologia baseada na técnica de Geração de Arcos para o Problema de Programação de Veículos**. 2001. 154f. Tese (Doutorado em Engenharia de Transportes) – Universidade de São Paulo, São Paulo, 2001.

SILVA, G. P; GUALDA, N. D. F. Um Algoritmo de Geração de Arcos para o Problema de Programação de Veículos. **Transportes**, v. 8, n.1, p. 35-54, 2001.