

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

FORMULAÇÕES E MÉTODOS DE SOLUÇÃO PARA O PROBLEMA DE COLETA
E ENTREGA COM JANELAS DE TEMPO, FROTA HETEROGÊNEA E
MÚLTIPLOS PRODUTOS

Diógenes Henrique Frazzato Gasque

São Carlos
Fevereiro de 2019

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

FORMULAÇÕES E MÉTODOS DE SOLUÇÃO PARA O PROBLEMA DE COLETA
E ENTREGA COM JANELAS DE TEMPO, FROTA HETEROGÊNEA E
MÚLTIPLOS PRODUTOS

Diógenes Henrique Frazzato Gasque

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção, da Universidade Federal de São Carlos como parte dos requisitos para obtenção do título de Mestre em Engenharia de Produção.

Orientador: Prof. Dr. Pedro Munari

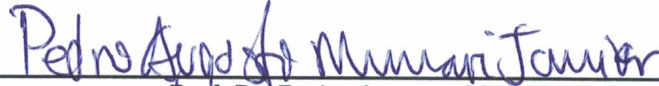
Financiamento: FAPESP (Processo nº 2016/24763-1) e CAPES

São Carlos
Fevereiro de 2019



Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Diógenes Henrique Frazzato Gasque, realizada em 21/02/2019:



Prof. Dr. Pedro Augusto Munari Junior

UFSCar



Prof. Dr. Reinaldo Morabito Neto

UFSCar



Profa. Dra. Franklina Maria Bragion de Toledo

USP

*A minha amada esposa Gilmara e
aos meus pais Marco e Sílvia.
Vocês foram minha motivação para que esta vitória fosse possível.*

Agradecimentos

A Deus por meu Senhor Jesus Cristo pois sem Ele nada seria possível, Ele é a fonte inspiradora de todo o saber.

A minha esposa Gilmara, que esteve comigo em todos os momentos dessa jornada, sempre me apoiou e motivou sem medir esforços.

Aos meus pais, Marco e Sílvia, que sempre me auxiliaram em tudo com todas as suas forças e pelo apoio sincero ao qual me deram.

Agradeço especialmente ao meu orientador, Prof. Dr. Pedro Munari, que com sabedoria me guiou até aqui, sempre mostrou-se pronto para sanar minhas dúvidas e em momentos difíceis, foi um amigo, com conselhos e palavras de encorajamento. Sou grato ao senhor professor, por ter me levado a ser um melhor pesquisador, profissional e por também ter aberto as portas me recebendo como seu aluno.

Aos professores da pós-graduação em especial ao Prof. Dr. Reinaldo Morabito, que com seu entusiasmo e conhecimento traz motivação a todos aqueles que se aventuram pela Pesquisa Operacional.

A todos meus amigos e colegas do Grupo de Pesquisa Operacional (GPO) da UFSCar.

A Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) processo nº 2016/24763-1 e a CAPES pelo apoio financeiro, sem o qual esta pesquisa não teria sido realizada.

*“Dedica grande consideração à sabedoria, e
ela te exaltarão; abraça-a, e ela te honrarão!”
Provérbios 4:8*

Resumo

A eficiência logística é fundamental para o sucesso dos negócios de grande parte das organizações. Inerente ao processo logístico, o roteamento de veículos desempenha papel importante na garantia de custos menores e satisfação dos clientes. O problema de roteamento de veículos com coleta e entrega surge em diversos contextos reais, tanto em ambientes urbanos como rurais, onde produtos são coletados em certos locais e transportados aos respectivos destinatários de entrega. Normalmente essas operações de coleta e entrega devem considerar decisões mais complexas, observando janelas de tempo de visita aos clientes, heterogeneidade da frota utilizada e quantidades a serem transportadas dos diversos produtos existentes. Este trabalho aborda o problema de roteamento de veículos com coleta e entrega e janelas de tempo, com restrições práticas de frota heterogênea, múltiplos produtos e fracionamento de carga. São propostos modelos e métodos de solução para o problema em questão. Um modelo de três-índices foi estendido com base na formulação clássica, contemplando todas as restrições práticas. Um segundo modelo, de dois-índices, foi adaptado para variantes contendo frota heterogênea e múltiplos produtos. Uma meta-heurística foi proposta e implementada para o caso clássico contendo frota homogênea e um único tipo de produto e suas variantes que abordam frota heterogênea, múltiplos produtos e coleta e entrega fracionada. Experimentos computacionais usando exemplares da literatura mostraram que os modelos e métodos propostos apresentaram resultados satisfatórios. Para facilitar o uso das abordagens propostas no apoio à tomada de decisão, desenvolveu-se uma ferramenta computacional com interface *web*, cujos resultados são apresentados nesse trabalho.

Palavras-chave: Roteamento de veículos, coleta e entrega, frota heterogênea, fracionamento de carga, modelos, meta-heurística.

Abstract

Logistics efficiency is critical to the business success of most organizations. Inherent in the logistics process, vehicle routing plays an important role to ensure lower costs and customer satisfaction. The vehicle routing problem with pickup and delivery arises in several real-world contexts in both urban and rural environments, where products are collected in certain locations and transported to the respective delivery recipients. Usually these pickup and delivery operations should consider more complex decisions, such as customer time windows, heterogeneity of the fleet, and quantities to be transported of the various existing products. We address the pickup and delivery vehicle routing problem with time windows, with practical constraints of heterogeneous fleet, multiple products and split load. We proposed models and solution methods for the problem in question. We extended a three-index model based on the classical formulation containing all practical constraints. We adapted a two-index model containing heterogeneous fleet and multiple products. Computational experiments using examples from literature showed that the models and methods proposed had acceptable results. We proposed a metaheuristic for the classical case and its variants. To facilitate the use of the proposed approaches for the support of decision-making, we developed a computational tool with web interface, and show the results of this tool.

Key words: Vehicle routing, pickup and delivery, heterogeneous fleet, split load, models, metaheuristics.

Sumário

| | | |
|----------|--|-----------|
| 1 | Introdução | 1 |
| 2 | Revisão da literatura | 4 |
| 2.1 | Problema de roteamento de veículos com janelas de tempo | 4 |
| 2.1.1 | Modelagem do PRVJT clássico | 4 |
| 2.1.2 | Métodos de solução para PRVJT clássico | 6 |
| 2.1.3 | Variantes do PRVJT: frota heterogênea e fracionamento de carga | 8 |
| 2.2 | Problema de coleta e entrega com janelas de tempo | 11 |
| 2.2.1 | Modelagem do PCEJT clássico | 11 |
| 2.2.2 | Métodos de solução para o PCEJT e variantes | 14 |
| 3 | Modelos para o PCEJT-1 e PCEJT-2 | 21 |
| 3.1 | Modelo de dois-índices para o PCJET-1 | 21 |
| 3.2 | Modelo de três-índices para o PCEJT-2 | 23 |
| 4 | Meta-Heurística para o PCEJT-1 e PCEJT-2 | 26 |
| 4.1 | Critério de aceite | 28 |
| 4.2 | Mecanismo adaptativo | 29 |
| 4.3 | Heurísticas de remoção | 30 |
| 4.3.1 | <i>Shaw Removal</i> e variantes | 30 |
| 4.3.2 | <i>Late Arrival Removal</i> | 31 |
| 4.3.3 | <i>Random Removal</i> | 31 |
| 4.3.4 | <i>Worst Removal</i> | 31 |
| 4.3.5 | <i>Random Route Removal</i> | 31 |
| 4.4 | Heurísticas de inserção | 32 |
| 4.4.1 | <i>Regret heuristics</i> | 33 |
| 4.4.2 | <i>Split heuristics</i> | 33 |
| 4.5 | Heurística construtiva | 34 |
| 5 | Resultados | 35 |
| 5.1 | Descrição das instâncias | 35 |
| 5.2 | Calibração dos parâmetros | 37 |
| 5.3 | Resultados da ALNS para o PCEJT clássico | 38 |
| 5.4 | Resultados para o PCEJT-1 | 42 |

| | | |
|----------|---|-----------|
| 5.5 | Resultados para o PCEJT-2 | 48 |
| 5.6 | Análise de Sensibilidade | 57 |
| 6 | <i>Software para web</i> | 66 |
| 6.1 | Tecnologia para o desenvolvimento | 66 |
| 6.2 | Documentação e funcionamento do <i>software</i> | 68 |
| 6.2.1 | Diagramas de casos de uso, classes e EER | 68 |
| 6.2.2 | Interfaces gráficas | 77 |
| 7 | Conclusões | 85 |
| | Referências | 87 |

Lista de Tabelas

| | | |
|----|---|----|
| 1 | Métodos de solução para o problema de coleta e entrega. | 20 |
| 2 | Características das instâncias de Ropke e Cordeau (2009). | 36 |
| 3 | Frota $F1$ de veículos heterogêneos para as classes de instâncias AA, BB, CC e DD. 36 | |
| 4 | Frota $F2$ de veículos heterogêneos para as classes de instâncias AA, BB, CC e DD. 37 | |
| 5 | Valores para os parâmetros da ALNS | 38 |
| 6 | Comparação entre resultados da meta-heurística e literatura. | 39 |
| 7 | Resultados para os modelos de três- e dois-índices. | 43 |
| 8 | Resultados para a meta-heurística. | 44 |
| 9 | Resultados para o modelo de três-índices com coleta e entrega fracionada. | 49 |
| 10 | Resultados para a meta-heurística com coleta e entrega fracionada. | 49 |
| 11 | Resultados modelo de três-índices com solução inicial. | 55 |
| 12 | Resultados análise de sensibilidade - Valores médios da função objetivo | 58 |
| 13 | Resultado análise de sensibilidade - Tempos médios para às soluções | 59 |
| 14 | Melhor Configuração por instância | 63 |
| 15 | Análise de sensibilidade para as heurísticas de inserção com ruído e sem ruído . . | 64 |
| 16 | Descrição das tabelas do diagrama do banco de dados. | 77 |

Lista de Figuras

| | | |
|----|---|----|
| 1 | Exemplo garantia de precedência e paridade (Furtado et al., 2017). | 13 |
| 2 | Grafo com nós reais e após a inserção dos nós artificiais. | 22 |
| 3 | Nova estrutura do grafo contendo nós artificiais. | 24 |
| 4 | Comparação entre soluções ALNS e Literatura para a classe AA. | 41 |
| 5 | Comparação entre soluções ALNS e Literatura para a classe BB. | 41 |
| 6 | Comparação entre soluções ALNS e Literatura para a classe CC. | 41 |
| 7 | Comparação entre soluções ALNS e Literatura para a classe DD. | 42 |
| 8 | Soluções para a classe AA. | 46 |
| 9 | Soluções para a classe BB. | 47 |
| 10 | Soluções para a classe CC. | 47 |
| 11 | Soluções para a classe DD. | 48 |
| 12 | Comparação entre ALNS e modelos de dois- e três-índices. | 48 |
| 13 | Soluções para a classe AA com coleta e entrega fracionada. | 52 |
| 14 | Soluções para a classe BB com coleta e entrega fracionada. | 53 |
| 15 | Soluções para a classe CC com coleta e entrega fracionada. | 53 |
| 16 | Soluções para a classe DD com coleta e entrega fracionada. | 54 |
| 17 | Comparação entre ALNS e modelo de três-índices. | 54 |
| 18 | Análise de sensibilidade dos tempos para instâncias com 15 requisições. | 61 |
| 19 | Análise de sensibilidade dos tempos para instâncias com 40 requisições. | 61 |
| 20 | Análise de sensibilidade dos tempos para instâncias com 65 requisições. | 61 |
| 21 | Análise de sensibilidade das heurísticas de inserção com e sem ruído. | 64 |
| 22 | Casos de uso. | 69 |
| 23 | Diagrama de classes de cliente | 70 |
| 24 | Diagrama de classes do plano de roteamento | 72 |
| 25 | Diagrama de classes. | 73 |
| 26 | Diagrama de classes. | 75 |
| 27 | Diagrama de banco de dados EER. | 76 |
| 28 | Tela de cadastro de clientes. | 78 |
| 29 | Tela de consulta de clientes. | 78 |
| 30 | Tela de gerenciamento de produtos. | 79 |
| 31 | Tela de gerenciamento de veículos. | 79 |
| 32 | Tela de montagem de requisições. | 80 |
| 33 | Tela de atribuição de demanda à requisição. | 80 |

| | | |
|----|--|----|
| 34 | Tela de montagem da frota de veículos a ser usada. | 81 |
| 35 | Tela de visualização das rotas geradas. | 82 |
| 36 | Tela de visualização da rota do veículo 2. | 82 |
| 37 | Tela de visualização da rota do veículo 1. | 83 |
| 38 | Tela de visualização da rota do veículo 3. | 83 |
| 39 | Tela de visualização da sequência de visitas para a rota do veículo 2. | 84 |

Capítulo 1

Introdução

O roteamento de veículos é uma atividade fundamental em Logística. Rotas eficientes permitem reduzir custos operacionais e emissão de gases, além de contribuir para a obtenção da qualidade de serviço desejada (Ballou, 2004). Dada a complexidade desta atividade e a necessidade de uma análise integrada, é importante que a tomada de decisões tenha o apoio de abordagens quantitativas, em particular aquelas oferecidas pela Pesquisa Operacional (PO).

Os métodos científicos usados em PO possibilitam representar matematicamente as situações encontradas na prática, analisá-las de forma rigorosa e obter soluções que sirvam como um suporte formal e confiável no processo de tomada de decisão (Shapiro, 2007; Arenales et al., 2007; Belfiore e Fávero, 2013). Nesse contexto, tem-se o estudo do Problema de Roteamento de Veículos (PRV), o qual tem contribuído de forma significativa para o tratamento de diversas situações práticas (Toth e Vigo, 2014; Golden et al., 2008). O uso de modelos matemáticos e algoritmos para este problema foram pioneiramente propostos por volta da década de 60 (Dantzig e Ramser, 1959; Balinski e Quandt, 1964; Clarke e Wright, 1964). Desde então, a literatura de PO abordando variantes do PRV e diferentes formas de solucioná-las têm sido bastante ativas (Laporte et al., 2013; Vidal et al., 2013b).

Apesar da importância prática e de uma literatura bastante ativa, observa-se um distanciamento entre pesquisa e indústria quando se diz respeito ao uso de métodos científicos no apoio à tomada de decisão, em particular no contexto de roteamento de veículos (Drexler, 2012; Xu et al., 2003). A dificuldade computacional de se resolver problemas de grande porte e a falta de uma comunicação efetiva entre pesquisadores e empresas são fatores que contribuem bastante para essa situação. Dessa forma, faz-se necessário que as pesquisas busquem o aprimoramento dos modelos matemáticos e dos métodos de solução, de modo que possam ser usados de forma ampla no apoio à tomada de decisão por parte das empresas. Além de eficazes em relação à velocidade de resolução dos problemas e qualidade das soluções obtidas, o desenvolvimento dessas abordagens deve incluir os tomadores de decisão e demais envolvidos no processo, para que características práticas relevantes sejam levadas em conta, resultando assim em decisões que possam ser de fato implantadas.

Buscando contribuir neste sentido, o presente trabalho aborda o PRV considerando requisitos práticos, motivados pela interação com a indústria. Em específico, o problema a ser considerado é uma variante do Problema de Coleta e Entrega com Janelas de Tempo (PCEJT) (Battarra

et al., 2014), no qual os clientes são particionados em fornecedores e consumidores e janelas de tempo são impostas para as visitas aos clientes. Além desses requisitos básicos, já considerados por abordagens da literatura, considerou-se características que são comuns na prática. Dessa forma, estudou-se duas variantes do PCEJT, a primeira contendo frota heterogênea e múltiplos produtos (PCEJT-1), e a segunda contendo as características anteriores mais o fracionamento da coleta e entrega (PCEJT-2). Tais características foram inspiradas na operação de uma cooperativa agrícola localizada no interior do Estado de São Paulo.

O estudo de operações de negócios envolvendo cooperativas agrícolas é relevante, pois um dos principais problemas enfrentados por produtores e cooperados do setor é relativo à distribuição de seus produtos. É notório que planejamento e infraestrutura inadequados causam ineficiência, o que culmina em custos operacionais maiores e perda de competitividade. Dessa forma, para comercialização de alimentos principalmente aqueles perecíveis como hortaliças e frutas, é necessário que os produtores rurais disponham de um sistema de distribuição eficiente que minimize suas perdas e custos ao longo da cadeia produtiva. Não obstante de fornecer alimentos com qualidade, é necessário fazê-lo com rapidez e cumprindo os prazos pré-estabelecidos dentro de janelas de tempo pequenas devido à própria natureza perecível do alimento. Além disso, fatores como frequência e agendamento ou horários da entrega dos produtos, foram destacados como importantes no estabelecimento de relações comerciais entre supermercados e hipermercados para a seleção de fornecedores de produtos hortícolas (Agustina et al., 2014; Ferneda et al., 2012). Outro ponto a ser observado é que a tomada de decisão em pequenas propriedades e cooperativas familiares no Brasil ainda é feita de forma empírica. Isso é decorrente da falta de recursos e acesso à informação e tecnologias existentes que tragam benefícios à administração do negócio (Ferneda et al., 2012).

Dessa forma, na tentativa de melhorar seus processos produtivos, cooperativas rurais e produtores familiares devem readequá-los, principalmente aqueles ligados à distribuição de mercadorias. Isso pode ser feito com o auxílio de ferramentas computacionais destinadas ao problema de roteamento de veículos (Toth e Vigo, 2014). No entanto, a maioria dessas ferramentas não se encaixa na estrutura organizacional, tão pouco leva em consideração as características distintas de uma pequena cooperativa agropecuária ou produtores rurais familiares. Caso esses grupos tenham acesso a métodos e ferramentas para auxílio à distribuição de seus produtos, seus custos poderiam ser reduzidos, a qualidade seria mantida ou até mesmo melhorada, haveria redução de perdas, impactos ambientais minimizados e o relacionamento com seus clientes poderia ser melhorado, provendo maior integração dos elos da cadeia produtiva em que estão inseridos (Ferneda et al., 2012).

Os objetivos deste trabalho foram: (i) propor modelos matemáticos para PCEJT-1 e PCEJT-2; (ii) desenvolver métodos de solução meta-heurísticos para os problemas; (iii) implementar um

software web de apoio à tomada de decisão fazendo uso dos modelos e dos métodos desenvolvidos. Essas variantes são bastante desafiadoras do ponto de vista computacional e têm sido pouco exploradas na literatura (Xu et al., 2003; Qu e Bard, 2013; Ríos-Mercado et al., 2013; Bettinelli et al., 2014). Além disso, possuem relevância prática ao modelar uma situação real enfrentada por várias organizações, em particular por cooperativas agrícolas, situação esta, que foi uma das inspirações destes trabalho. Não foi encontrado na literatura nenhum trabalho que incorpore todas as características observadas na prática para o caso em estudo. Diferentes modelos e métodos de solução, exatos e heurísticos, para a variante clássica do PCEJT com mais de um veículo foram propostos na literatura (Battarra et al., 2014), mas a maioria dos trabalhos aborda variantes com frota homogênea e um único produto, o que dificulta o uso dessas abordagens na prática. Dessa forma, as abordagens propostas contribuem com o estado-da-arte e resultaram em uma ferramenta computacional com o papel de auxiliar a tomada de decisão em situações reais além de poder servir como *interface* gráfica para outras implementações que abordem o problema de roteamento de veículos.

Este trabalho está dentro do contexto da Pesquisa Operacional, no qual se faz uso de modelos matemáticos e de técnicas de programação para abstração e resolução de um problema. Tendo esta pesquisa o intuito de propor novos modelos e melhorias para os já existentes, bem como métodos de resolução para um problema já idealizado na literatura, como é o caso do problema de roteamento de veículos, pode-se considerar então o método de pesquisa deste trabalho como *Axiomático Normativo*. Contudo, devido à inspiração da pesquisa ser oriunda de uma situação real, com possível aplicação futura dos métodos e de modelos resultantes para proposição de melhorias, também existe o carácter *Empírico Normativo* do método de pesquisa (Morabito e Pureza, 2012; Bertrand e Fransoo, 2002).

As demais seções deste trabalho estão organizadas da seguinte forma. No Capítulo 2, define-se o problema a ser estudado juntamente com uma revisão da literatura e caracterização do caso prático considerado. Nos Capítulos 3 e 4, são apresentados os modelos e a meta-heurística propostas neste trabalho. No Capítulo 5, são apresentadas as características das instâncias usadas e os resultados dos testes computacionais para a variante clássica do problema de coleta e entrega com janelas de tempo (PCEJT) e para os casos com frota heterogênea, múltiplos produtos e coleta e entrega fracionada. No Capítulo 6, são descritas a tecnologia usada, a documentação e interfaces do *software*. Por fim, no Capítulo 7, são apresentadas as conclusões desta pesquisa respectivamente.

Capítulo 2

Revisão da literatura

Neste capítulo, apresenta-se a revisão bibliográfica dos problemas estudados, iniciando-se pelo problema de roteamento de veículos com janela de tempo (PRVJT). Em seguida, apresenta-se uma formulação da variante clássica do PCEJT e discute-se sobre métodos de solução para as variantes básicas e práticas do problema.

2.1 Problema de roteamento de veículos com janelas de tempo

O PRVJT tem sido usado com sucesso na modelagem de situações reais de grande importância no contexto industrial. A todo momento, diversas empresas estão buscando as melhores rotas para atender seus clientes por meio da entrega e/ou coleta de produtos. Alguns exemplos interessantes da aplicação do PRVJT podem ser encontradas em, por exemplo, Toth e Vigo (2014); Golden et al. (2008); Braysy e Gendreau (2005).

2.1.1 Modelagem do PRVJT clássico

No PRVJT clássico, uma frota de veículos está disponível em um único depósito e esta frota deve ser usada para visitar n clientes de modo a atender as demandas d_i de cada cliente $i = 1, \dots, n$. O objetivo principal é determinar rotas que tenham custo total mínimo e satisfaçam os seguintes requisitos. Cada cliente deve ser visitado exatamente uma única vez e a visita deve satisfazer a janela de tempo $[w_i^a, w_i^b]$, isto é, o veículo não pode chegar no cliente após o instante w_i^b e deve esperar até w_i^a para iniciar o serviço caso chegue antes deste instante. O tempo de serviço necessário para atender um cliente i é fixo e denotado por s_i . O tempo de viagem entre dois clientes i e j é dado por t_{ij} . O custo de viagem entre dois nós i e j é dado por c_{ij} , sendo $c_{ij} = 0$ para $i = j$. Os veículos devem retornar ao depósito após realizarem as rotas designadas. Cada veículo tem uma capacidade máxima q , o que limita o número de clientes que ele pode visitar. Assume-se que os veículos sejam idênticos e que haja um número suficiente de veículos para atender a demanda de todos os clientes. Para uma descrição detalhada do problema, recomendam-se os trabalhos de Kallehauge et al. (2005) e Feillet (2010).

Seja K o número de veículos disponíveis no depósito e sejam 0 e $n + 1$ os índices usados para representar o depósito. Esta notação é usada para diferenciar quando um veículo deixa o depósito (índice 0) de quando o veículo retorna ao depósito (índice $n + 1$). Assim, os clientes e o depósito ficam representados por um grafo completo contendo $n + 2$ nós, numerados de 0 a

$n + 1$. O PRVJT pode ser modelado de acordo com a seguinte formulação:

$$\min \quad \sum_{k=1}^K \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ijk} \quad (2.1a)$$

$$\text{s.a} \quad \sum_{k=1}^K \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ijk} = 1, \quad i = 1, \dots, n, \quad (2.1b)$$

$$\sum_{i=1}^n d_i \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ijk} \leq q, \quad k = 1, \dots, K, \quad (2.1c)$$

$$\sum_{j=1}^{n+1} x_{0jk} = 1, \quad k = 1, \dots, K, \quad (2.1d)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^n x_{ihk} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hjk} = 0, \quad h = 1, \dots, n, \quad k = 1, \dots, K, \quad (2.1e)$$

$$\sum_{i=0}^n x_{i,n+1,k} = 1, \quad k = 1, \dots, K, \quad (2.1f)$$

$$w_{ik} + (s_i + t_{ij})x_{ijk} - M_{ij}(1 - x_{ijk}) \leq w_{jk}, \quad i, j = 0, \dots, n + 1, \quad k = 1, \dots, K, \quad (2.1g)$$

$$w_i^a \leq w_{ik} \leq w_i^b, \quad i = 0, \dots, n + 1, \quad k = 1, \dots, K, \quad (2.1h)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j = 0, \dots, n + 1, \quad k = 1, \dots, K, \quad (2.1i)$$

sendo x_{ijk} uma variável de decisão que assume valor 1 se o veículo k visita o nó i e vai imediatamente para o nó j em seguida; caso contrário, tem-se $x_{ijk} = 0$. A variável de decisão w_{ik} determina o instante de tempo a partir do qual o veículo k pode iniciar o serviço no cliente i (este valor é irrelevante quando o cliente i não é visitado pelo veículo k). A função objetivo (2.1a) consiste em minimizar os custos de roteamento. As restrições (2.1b) garantem que cada cliente seja visitado por um único veículo, enquanto as restrições (2.1c) evitam que um veículo exceda sua capacidade máxima. As restrições (2.1d) e (2.1f) garantem que cada veículo deixe o depósito uma única vez e também retorne uma única vez. As restrições (2.1e) garantem que, após visitar um cliente, o veículo deve ir para um outro cliente ou retornar para o depósito. A satisfação das janelas de tempo são garantidas pelas restrições (2.1g) e (2.1h), sendo que (2.1g) também evitam a formação de sub-rotas. O parâmetro $M_{ij} = \max\{w_i^b - w_j^a, 0\}$ é usado para tornar inativa a restrição correspondente sempre que a variável x_{ijk} assumir valor zero. Observe que quando um veículo k não é utilizado, tem-se $x_{0,n+1,k} = 1$.

No modelo (2.1), os veículos são considerados de forma explícita, já que a variável de decisão é também indexada por $k = 1, \dots, K$. Entretanto, ao assumir que a frota de veículos é homogênea (isto é, os veículos são idênticos), não há a necessidade de se explicitar qual veículo será utilizado para executar uma dada rota. Basta que a variável de decisão determine as rotas a serem usadas para atender a demanda de todos os clientes. Dessa forma, uma formulação com um menor

número de variáveis é obtida:

$$\min \quad \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} c_{ij} x_{ij} \quad (2.2a)$$

$$\text{s.a} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ij} = 1, \quad i = 1, \dots, n, \quad (2.2b)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^n x_{ih} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hj} = 0, \quad h = 1, \dots, n, \quad (2.2c)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^n x_{ij} = 1, \quad j = 1, \dots, n, \quad (2.2d)$$

$$y_i + d_j x_{ij} - q(1 - x_{ij}) \leq y_j, \quad i, j = 0, \dots, n + 1, \quad (2.2e)$$

$$w_i + (s_i + t_{ij})x_{ij} - M_{ij}(1 - x_{ij}) \leq w_j, \quad i, j = 0, \dots, n + 1, \quad (2.2f)$$

$$d_i \leq y_i \leq q, \quad i = 0, \dots, n + 1, \quad (2.2g)$$

$$w_i^a \leq w_i \leq w_i^b, \quad i, j = 0, \dots, n + 1, \quad (2.2h)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, n + 1. \quad (2.2i)$$

Nesta formulação, a variável de decisão x_{ij} assume valor 1 quando existe uma rota que visita o nó i e segue imediatamente para o nó j . A variável de decisão w_i determina o instante de tempo em que o serviço no nó i é iniciado. Observe que as variáveis auxiliares y_i foram introduzidas no modelo, sendo que cada y_i determina a demanda total já atendida pela rota ao visitar o nó i , para $i = 0, \dots, n + 1$, sendo que $d_0 = 0$ e $d_{n+1} = 0$. Assim, as restrições (2.2e) e (2.2g) juntas garantem que a capacidade de cada veículo é respeitada. Os demais parâmetros e restrições são similares aos descritos em (2.1).

2.1.2 Métodos de solução para PRVJT clássico

Compreendendo a importância do PRV para diversos tipos de problemas logísticos (Toth e Vigo, 2014; Vidal et al., 2013a), faz-se necessário também apresentar alguns métodos de solução eficientes para o PRVJT. Sendo assim, apresenta-se primeiramente três abordagens heurísticas e uma abordagem exata para o PRVJT e algumas variantes. Todas as abordagens mencionadas são conhecidas por serem o estado-da-arte para métodos de solução para o PRVJT.

Nagata et al. (2010) propuseram um Algoritmo Memético (MA, do inglês *memetic algorithm*) para o PRVJT clássico. Abordagens do tipo MA são caracterizadas por serem heurísticas baseadas em população, combinando um algoritmo evolucionário para diversificação com outro de busca local para intensificação. Dessa forma, MAs também podem ser referidos como algoritmos genéticos híbridos. O MA sugerido pelos autores tem duas características importantes, a primeira é a Adaptação do Cruzamento de Montagens de Arestas (EAX, do inglês *edge assembly*

crossover) do PRV para o PRVJT. A segunda é a introdução de uma nova função de custo penalizada para a violação de janelas de tempo. Outra característica apresentada pelos autores é a divisão do algoritmo em duas fases, onde primeiramente, uma população inicial contendo um mesmo número de rotas minimizado para cada solução é obtido. Em seguida, utiliza-se um procedimento para minimizar as distâncias percorridas do número de rotas determinado no passo anterior. Os resultados computacionais da abordagem proposta usando instâncias da literatura evidenciaram sua competitividade frente a outras abordagens, dado que ela permitiu obter 186 novas melhores soluções de 356 instâncias em tempos computacionais razoáveis.

Vidal et al. (2013a) abordaram o PRVJT clássico bem como outras variantes contendo múltiplos depósitos, múltiplos períodos e com veículos dependentes de locais. Os autores propuseram uma meta-heurística híbrida denominada Busca Genética Híbrida com Controle de Diversidade Avançado (HGSADC, do inglês *Hybrid Genetic Search with Advanced Diversity Control*). Essa abordagem é uma junção das propriedades de exploração do algoritmo genético com procedimentos de busca local e mecanismos de gerenciamento de diversificação. Uma característica importante da HGSADC é que a diversidade da população é um objetivo que também deve ser otimizado juntamente com a qualidade da solução. Os autores de maneira geral apresentaram um conceito generalizado para a HGSADC envolvendo classes diferentes do PRVJT contendo características tanto temporais como geométricas. Também introduziram novos procedimentos de buscas em vizinhanças considerando soluções ineficazes com relação à duração de rotas e janelas de tempos. Outro ponto importante abordado pelos autores foi o estabelecimento de princípios para decomposição de grandes instâncias tornando-as eficientemente tratáveis do ponto de vista da HGSADC. Os resultados computacionais para a meta-heurística proposta demonstraram que a mesma superou outras abordagens existentes na literatura em relação à qualidade das soluções e também em eficiência computacional, isso para quatro classes de problemas de roteamento de veículos contemplados.

Christiaens e Vanden Berghe (2018) propuseram uma abordagem heurística baseada no princípio de destruição e reconstrução (Schrimpf et al., 2000) juntamente com o critério de aceite do Recozimento Simulado (SA, do inglês *Simulated Annealing*) (Kirkpatrick et al., 1983) para o PRVJT e outras variantes como, por exemplo, o PRV com múltiplos depósitos. Os autores referem-se ao método como Indução de Folgas por Remoção de Linhas (SISRs, do inglês *Slack Induction by String Removals*). O método é composto por um primeiro procedimento de destruição que tenta remover uma sequência de clientes suficientemente grande formando uma linha, ou seja, clientes adjacentes um ao outro. Dessa forma, espera-se criar folgas nas soluções, estas folgas são referidas como sendo de capacidade e espaciais, podendo esta última ser vista como se o alcance do veículo fosse limitado. O segundo procedimento é o de reconstrução que reinsere os clientes removidos anteriormente. Este procedimento de inserção é um algoritmo

guloso com “piscadas”, que na verdade é uma forma de avaliar todas as posições em uma rota com uma determinada probabilidade, caso contrário o algoritmo “pisca”, ou seja, pula aquela posição. Antes da inserção, os clientes são ordenados de acordo com algum critério, sendo que os autores propuseram ordená-los de forma aleatória, de acordo com a demanda, se estão mais perto ou mais longe do depósito. Qualquer uma destas opções de ordenamento podem ser selecionados aleatoriamente. O último procedimento é referente à minimização de veículos, que é aplicada inicialmente durante 10% do tempo de busca. A fase de minimização de veículos utiliza os mecanismos de destruição e reconstrução mudando-se somente o critério de aceite da SA, que passa levar em conta se um cliente está ou não na solução. Os resultados computacionais para instâncias da literatura do PRVC demonstraram que a SISRs teve bons desempenhos para instâncias de pequeno e médio porte. Já para instâncias de grande porte, o método proposto superou outras abordagens apresentadas na literatura. Para outras variantes do PRV, a heurística SISRs mostrou-se competitiva em relação a outros métodos até então propostos, igualando ou até mesmo melhorando as melhores soluções conhecidas até então para instâncias da literatura.

Pecin et al. (2017) propuseram um algoritmo do tipo *branch-and-price-and-cut* (BPC) para o PRVJT clássico. Baseado na formulação por particionamento de conjuntos do problema (Desaulniers et al., 2014), os autores adicionaram duas melhorias no algoritmo BPC até então estado-da-arte na literatura, sendo que na primeira utilizou-se desigualdades válidas denominadas cortes de subconjuntos de linhas com limite de memória (Jepsen et al., 2008). Para estas desigualdades os autores definiram a memória em termos do subconjunto de arcos ao invés do subconjunto de nós. Dessa forma, foi possível atenuar o impacto negativo no tempo computacional que as variáveis duais destes cortes infligem no algoritmo de rótulos. A segunda melhoria é referente ao uso de um conjunto novo de desigualdades válidas derivadas do conjunto proposto por Balas (1977). Além destas duas melhorias propostas, o algoritmo de rótulos apresentado pelos autores implementa várias técnicas de aceleração, tais como: busca bidirecional (*bidirectional search*), complementação de limitantes (*completion bounds*), relaxação de estado-espaco decremental (*decremental state space relaxation*) e heurística de rotulagem. O algoritmo BPC foi testado em dois conjuntos de instâncias da literatura. No primeiro, composto de instâncias com 100 clientes, o algoritmo conseguiu prover soluções e provar otimalidade para as 56 instâncias. Para o segundo conjunto de instâncias, com 200 clientes, o algoritmo BPC conseguiu prover soluções para 60 instâncias provando otimalidade para 51 delas.

2.1.3 Variantes do PRVJT: frota heterogênea e fracionamento de carga

Após a apresentação de alguns métodos de solução para o PRV e algumas de suas variantes é passível de menção algumas outras, sendo que algumas delas surgem naturalmente motivadas por aplicações de problemas reais (Toth e Vigo, 2014; Golden et al., 2008). Dentre elas, pode-se

citar o PRV com frota heterogênea e o PRV com fracionamento de carga. Essas duas variantes são estudadas neste trabalho, tornando pertinente a descrição das alterações impostas ao modelo clássico da literatura (2.1).

De acordo com Golden et al. (2008) e Toth e Vigo (2014), a característica heterogênea da frota pode estar na capacidade dos veículos, nos custos fixos e também nos variáveis. Dependendo do problema uma ou mais características podem ser heterogêneas. Dessa forma, assume-se uma frota de K veículos disponível no depósito, de modo que cada veículo $k = 1, \dots, K$ tem capacidade igual a q_k . Analogamente à capacidade dos veículos, tem-se o custo fixo associado ao uso de cada veículo do tipo k , sendo igual a f_k . Seguindo o mesmo raciocínio, têm-se custos variáveis diferentes associados a cada veículo tipo k . Para um mesmo trajeto, por exemplo, quando parte-se de um cliente i para j , os custos variam de acordo com o veículo usado. Sendo assim um custo não negativo, c_{ijk} , é associado a cada tipo de veículo k . Dentro do contexto desta pesquisa, considera-se somente a heterogeneidade da capacidade dos veículos da frota

No presente trabalho, faz-se o uso de veículos com diferentes capacidades considerando-se uma frota limitada. Assim, as adaptações apresentadas são referentes à essa característica. Para a formulação clássica de três-índices é necessário adaptar somente as restrições (2.1c) que passam a ser expressas por:

$$\sum_{i=1}^n d_i \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ijk} \leq q_k, \quad k = 1, \dots, K. \quad (2.3a)$$

Yaman (2006) apresenta seis novas formulações para o PRVC com frota heterogênea, no entanto, apresenta-se somente uma delas para ilustrar as alterações impostas para a formulação de dois-índices. Sabe-se também que para o contexto do PRV, existe a possibilidade de considerar-se veículos com diferentes capacidades lançando mão de nós artificiais (Ropke et al., 2007). No entanto, a intenção aqui é apenas demonstrar a existência e a possibilidade de usar-se de uma frota heterogênea em uma formulação de dois-índices para o problema de roteamento de veículos capacitado. O emprego de nós artificiais para a inclusão de frota heterogênea será discutido na Seção 3.1. Para a formulação de dois-índices do PRVC Yaman (2006) propõem adicionar uma variável binária adicional b_{ik} , que assume valor 1 caso haja uma rota k em que o nó i é o último visitado. Faz-se necessário assumir que a demanda d_i para todo $i = 1, \dots, n$, pode ser transportada pelo veículo da frota com maior capacidade, caso contrário o problema é infactível. Diante dessa afirmação, e sem perda de generalidade assume-se que $q_1 \geq q_2, \geq \dots \geq q_{|K|}$ e sendo assim, pode-se realizar a substituição de M_{ij} por q_1 , como sendo um valor suficientemente grande (Yaman, 2006). A formulação de dois-índices adaptada para frota heterogênea deve ter a restrição (2.2g) substituída por (2.4c) e (2.4d). Além disso, é necessário acrescentar as restrições (2.4a), (2.4b) e (2.4e).

$$\sum_{j=1}^n x_{0j} = \sum_{i=1}^n \sum_{k=1}^K b_{ik}, \quad (2.4a)$$

$$\sum_{\substack{j=0 \\ j \neq i}}^n x_{ij} = 1 - \sum_{k=1}^K b_{ik}, \quad i = 1, \dots, n, \quad (2.4b)$$

$$y_i \geq d_i + \sum_{\substack{j=0 \\ j \neq i}}^n d_j \cdot x_{ji}, \quad i = 1, \dots, n, \quad (2.4c)$$

$$y_i \leq \sum_{k=1}^K q_k \cdot b_{ik} + q_1 \left(1 - \sum_{k=1}^K b_{ik}\right), \quad i = 1, \dots, n, \quad (2.4d)$$

$$b_{ik} \in \{0, 1\}, \quad i = 0, \dots, n; k = 1, \dots, K. \quad (2.4e)$$

As restrições (2.4a) garantem que o número de rotas seja igual ao número de vezes em que o depósito inicial é deixado. As restrições (2.4b) asseguram que haja um veículo k deixando o nó i caso a rota realizada pelo veículo k visite o nó i por último. A soma da carga acumulada após visitar-se um nó i para cada rota é feita pelas restrições (2.4c) juntamente com (2.2e) da formulação (2.2). Para assegurar que a capacidade dos veículos seja respeitada impõem-se as restrições (2.4d).

A motivação para o emprego do fracionamento de carga é referente às reduções dos custos de roteamento e do número de rotas quando comparadas com soluções que não permitem o fracionamento. Também é possível lidar com demandas maiores que a capacidade dos veículos utilizados. A formulação de três-índices (2.1) para o PRV restringe as visitas aos clientes a somente, ou seja, um cliente só ser visitado por um único veículo. Enquanto que para a formulação com fracionamento essa restrição deve ser relaxada, isso faz que um cliente possa ser visitado mais de uma vez desde que seja por veículos distintos. No contexto desta pesquisa, não é permitido que um cliente seja visitado mais de uma vez por um mesmo veículo. Uma variável $y_{ik} \geq 0$ é necessária para representar a quantidade entregue a um cliente i pelo veículo k (Archetti e Speranza, 2012; Toth e Vigo, 2014). Dessa forma, tem-se:

$$\sum_{k=1}^K \sum_{\substack{j=1 \\ j \neq i}}^{n+1} x_{ijk} \geq 1, \quad i = 1, \dots, n, \quad (2.5a)$$

$$\sum_{i=1}^n y_{ik} \leq q_k, \quad k = 1, \dots, K, \quad (2.5b)$$

$$y_{ik} \leq d_i \sum_{\substack{j=0 \\ j \neq i}}^n x_{ijk} \quad i = 1, \dots, n; k = 1, \dots, K, \quad (2.5c)$$

$$\sum_{k=1}^K y_{ik} = d_i \quad i = 1, \dots, n. \quad (2.5d)$$

As restrições (2.1b) e (2.1c) foram reescritas e devem ser substituídas por (2.5a) e (2.5b) respectivamente. Elas garantem que cada cliente seja visitado ao menos uma vez e asseguram que a capacidade dos veículos não seja excedida. As restrições (2.5c) asseguram que quando um veículo visita um nó i , ele entregue uma quantidade menor ou igual à demanda daquele cliente i . Por fim, as restrições (2.5d) asseguram que a demanda de cada cliente seja atendida por um ou mais veículos.

Outra característica abordada nesta pesquisa é a necessidade de transportar-se múltiplos produtos. Ao se diferenciar os produtos, remete-se principalmente a diferença da qualidade que pode existir entre eles e se os mesmos podem ser misturados ou não. Alguns exemplos da necessidade de explicitar os diferentes produtos aparecem dentro do contexto de transporte de óleo de oliva (Lahyani et al., 2015), leite cru (Paredes-Belmar et al., 2016), produtos derivados do petróleo (Cornillier et al., 2008) e na coleta de lixo seletiva (Henke et al., 2015). Em um contexto diferente, onde aborda-se problemas de roteamento de estoques (IRPs, do inglês *Inventory Routing Problems*), pode-se citar aqueles oriundos da indústria de bebidas e supermercados, onde além de diferenças de qualidade, outros fatores como o nível de estoque de cada produto são levados em consideração (Coelho e Laporte, 2013).

2.2 Problema de coleta e entrega com janelas de tempo

O PCEJT é uma variante do PRVJT apresentado anteriormente, na qual o atendimento de uma dada demanda exige visitar dois clientes: um para a coleta e outro para a entrega do produto. Cada par de clientes associado a uma requisição de coleta e entrega deve ser visitado pelo mesmo veículo, garantindo-se que a visita ao cliente de coleta preceda a visita ao cliente de entrega (não necessariamente de forma imediata). Diversas situações práticas podem ser modeladas pelo PCEJT, incluindo serviços de entrega urbana, transporte marítimo de petróleo, transporte escolar, transporte de animais e deslocamento de pessoas idosas e deficientes sob demanda (Furtado et al., 2015; Battarra et al., 2014; Cordeau e Laporte, 2007; Sigurd et al., 2004; Qu e Bard, 2013).

2.2.1 Modelagem do PCEJT clássico

Uma formulação para o PCEJT pode ser obtida como uma extensão da formulação (2.1) em que cada demanda d_i está associada a uma requisição de coleta no cliente i e entrega no cliente $n + i$, $i = 1, \dots, n$. Assim, clientes de coleta são representados pelos nós de 1 a n , enquanto clientes de entrega pelos nós de $n + 1$ a $2n$. O depósito fica representado pelos nós 0 e $2n + 1$. Assume-se que a demanda do cliente de coleta i é dada por d_i , enquanto que a demanda do cliente de entrega é dada por $d_{n+i} = -d_i$. Esta definição permite que o balanceamento de carga dos veículos seja feito por restrições similares a (2.2e), usada na formulação do PRVJT. Usando

essas definições, uma formulação do PCEJT é dada por:

$$\min \quad \sum_{k=1}^K \sum_{i=0}^{2n+1} \sum_{j=0}^{2n+1} c_{ij} x_{ijk} \quad (2.6a)$$

$$\text{s.a} \quad \sum_{k=1}^K \sum_{\substack{j=1 \\ j \neq i}}^{2n+1} x_{ijk} = 1, \quad i = 1, \dots, n, \quad (2.6b)$$

$$\sum_{j=1}^{2n+1} x_{0jk} = 1, \quad k = 1, \dots, K, \quad (2.6c)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^{2n} x_{ihk} - \sum_{\substack{j=1 \\ j \neq h}}^{2n+1} x_{hjk} = 0, \quad h = 1, \dots, 2n, \quad k = 1, \dots, K, \quad (2.6d)$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{2n} x_{ijk} - \sum_{\substack{i=1 \\ i \neq j+n}}^{2n} x_{i,j+n,k} = 0, \quad j = 1, \dots, n, \quad k = 1, \dots, K, \quad (2.6e)$$

$$\sum_{i=0}^{2n} x_{i,2n+1,k} = 1, \quad k = 1, \dots, K, \quad (2.6f)$$

$$y_{ik} + d_j x_{ijk} - q(1 - x_{ijk}) \leq y_{jk}, \quad i, j = 0, \dots, 2n+1, \quad k = 1, \dots, K, \quad (2.6g)$$

$$w_{ik} + (s_i + t_{ij})x_{ijk} - M_{ij}(1 - x_{ijk}) \leq w_{jk}, \quad i, j = 0, \dots, 2n+1, \quad k = 1, \dots, K, \quad (2.6h)$$

$$w_{ik} + s_i + t_{i,n+i} \leq w_{n+i,k}, \quad i = 1, \dots, 2n, \quad k = 1, \dots, K, \quad (2.6i)$$

$$w_i^a \leq w_{ik} \leq w_i^b, \quad i = 0, \dots, 2n+1, \quad k = 1, \dots, K, \quad (2.6j)$$

$$\max\{0, d_i\} \leq y_{ik} \leq \min\{q, q + d_i\}, \quad i = 0, \dots, 2n+1, \quad k = 1, \dots, K, \quad (2.6k)$$

$$x_{ijk} \in \{0, 1\}, \quad i, j = 0, \dots, 2n+1, \quad k = 1, \dots, K. \quad (2.6l)$$

As variáveis de decisão, os parâmetros e as restrições seguem a mesma interpretação apresentada para o PRVJT. As restrições (2.6e) e (2.6i) são adicionadas ao modelo para impor respectivamente que os nós de coleta e de entrega de uma mesma requisição sejam visitados pelo mesmo veículo (pareamento) e que o instante de tempo da visita ao nó de entrega $n + i$ seja posterior ao instante da visita ao nó de coleta i (precedência).

Uma formulação compacta de dois-índices para o PCEJT clássico é apresentada por Furtado et al. (2017). No trabalho, os autores propuseram uma nova maneira de garantir a precedência e paridade das visitas aos nós de coleta e de entrega. A garantia é feita pela inserção de novas variáveis contínuas, v_i , para todos os nós de coleta e de entrega. Essas variáveis são responsáveis por armazenar o índice do primeiro nó de uma rota e realizar a propagação desse valor para todos os nós pertencentes a essa mesma rota. Na Figura 1, representa-se duas rotas, r_1 e r_2 , e as variáveis v_i com seus respectivos valores. Os depósitos inicial e final são representados por 0 e $2n + 1$. Os nós de coleta são representados de 1 a 3, os nós de entrega de 4 a 6. Observe que $v_1 = v_3 = v_4 = v_6 = 1$, dado que o nó 1 é o primeiro nó visitado por r_1 e que os nós 3, 4 e 6 também pertencem a r_1 . Analogamente, tem-se $v_2 = v_5 = 1$, já que o nó 2 é o primeiro nó visitado por r_2 . Com isto, é possível garantir o pareamento dos nós de coleta e entrega, impondo-se que $v_{n+i} = v_i$, para todo i, \dots, n .

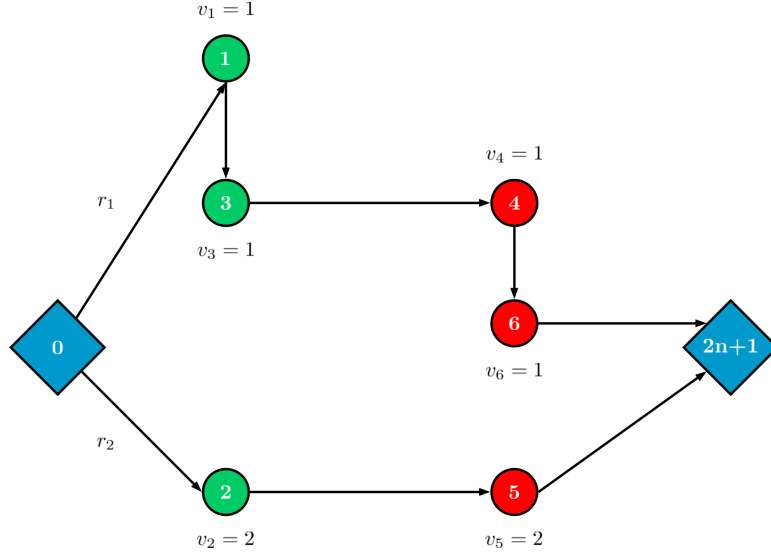


Figura 1: Exemplo garantia de precedência e paridade (Furtado et al., 2017).

Devido a essa nova estratégia de garantia de precedência e de paridade, o desempenho da nova formulação foi superior ao modelo clássico de três-índices e ao modelo de dois-índices apresentado por Lu e Dessouky (2004). De um total de 60 instâncias, a formulação de Furtado et al. (2017) resolveu 26 instâncias até a otimalidade, a formulação clássica de três-índices conseguiu resolver 15 instâncias enquanto a formulação de Lu e Dessouky (2004) resolveu apenas 10. Além disso, os tempos computacionais foram significativamente menores. A formulação de dois-índices é apresentada a seguir:

$$\min \quad \sum_{i=0}^{2n+1} \sum_{j=0}^{2n+1} c_{ij} x_{ij} \quad (2.7a)$$

$$\text{s.a} \quad \sum_{\substack{i=0 \\ i \neq j}}^{2n} x_{ij} = 1, \quad j = 1, \dots, 2n+1, \quad (2.7b)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{2n+1} x_{ij} = 1, \quad i = 0, \dots, 2n, \quad (2.7c)$$

$$w_j \geq w_i + t_{ij} - M(1 - x_{ij}), \quad i, j = 0, \dots, 2n+1, \quad (2.7d)$$

$$w_i^a \leq w_i \leq w_i^b, \quad i = 0, \dots, 2n+1, \quad (2.7e)$$

$$y_j \geq y_i + d_j - M(1 - x_{ij}), \quad i, j = 0, \dots, 2n+1, \quad (2.7f)$$

$$\max\{0, d_i\} \leq y_i \leq \min\{q, q + d_i\}, \quad i = 0, \dots, 2n+1, \quad (2.7g)$$

$$w_{n+i} \geq w_i + s_i + t_{i,i+n}, \quad i = 1, \dots, n, \quad (2.7h)$$

$$v_{n+i} = v_i, \quad i = 1, \dots, n, \quad (2.7i)$$

$$v_j \geq j \cdot x_{0j}, \quad j = 1, \dots, 2n, \quad (2.7j)$$

$$v_j \leq j \cdot x_{0j} - n(x_{0j} - 1), \quad j = 1, \dots, 2n, \quad (2.7k)$$

$$v_j \geq v_i + n(x_{ij} - 1), \quad i, j = 1 \dots, 2n, \quad (2.7l)$$

$$v_j \leq v_i + n(1 - x_{ij}), \quad i, j = 1 \dots, 2n, \quad (2.7m)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, 2n + 1. \quad (2.7n)$$

A função objetivo (2.7a) consiste em minimizar os custos de roteamento. As restrições (2.7b) e (2.7c) asseguram que os clientes sejam visitados somente uma única vez. A progressividade dos tempos de visita aos clientes é garantida por (2.7d), as quais também evitam sub-rotas. O respeito às janelas de tempo é imposto pelas restrições (2.7e). O fluxo de carga dos veículos é assegurado pelas restrições (2.7f). As restrições (2.7g) garantem que a capacidade dos veículos não seja ultrapassada. A precedência e paridade das visitas aos clientes é garantida pelas restrições (2.7h) a (2.7m). A garantia que o tempo de visita ao nó de entrega seja feita em momento posterior ao respectivo nó de coleta, é imposta pelas restrições (2.7h). As restrições (2.7i) garantem que o nó de coleta e de entrega de uma requisição tenham o mesmo identificador. A captura do índice identificador de uma rota é feito pelas restrições (2.7j). As restrições (2.7k) garantem que o primeiro identificador de uma rota seja o índice do primeiro nó ou zero. As restrições (2.7l) e (2.7m) propagam o identificador aos demais nós de uma mesma rota. Por fim, as restrições (2.7n) garantem a integralidade das variáveis de decisão.

Outras formulações estão propostas na literatura para o PCEJT, conforme revisado por Battarra et al. (2014) e Furtado et al. (2017). Algumas delas são definidas por um número de restrições que é exponencial em relação ao número de requisições de coleta e de entrega no problema e, assim, requerem a implementação de um método *branch-and-cut* específico para sua solução (Lu e Dessouky, 2004; Ropke et al., 2007). Formulações do tipo *particionamento de conjuntos* também estão propostas e exigem a implementação de um método *branch-and-price* (Ropke e Cordeau, 2009; Baldacci et al., 2011), sendo atualmente a abordagem mais efetiva para a obtenção de soluções ótimas, embora os tempos computacionais ainda sejam bastante elevados em problemas de grande porte. Métodos heurísticos têm sido propostos para o PCEJT (Lu e Dessouky, 2006; Ropke e Pisinger, 2006) e, de modo geral, permitem obter soluções de boa qualidade, inclusive para exemplares de grande porte. Em todos esses trabalhos citados, considera-se frota homogênea, único tipo de produto e não fracionamento de carga.

2.2.2 Métodos de solução para o PCEJT e variantes

Nesta seção, apresenta-se uma revisão dos métodos de solução para o PCEJT. Primeiramente, revisam-se os trabalhos com abordagens heurísticas e meta-heurísticas, e posteriormente, trabalhos com abordagens exatas.

Li e Lim (2001) propuseram uma meta-heurística que combina Busca Tabu (TS, do inglês *Tabu Search*) e SA para o PCEJT. Na TS proposta, as soluções visitadas são armazenadas em uma lista tabu evitando repetições de movimentos. Já para a implementação da SA há uma

diferença da implementação proposta pelos autores para as usuais, que normalmente determinam o aceite de uma solução conforme uma medida probabilística. A mudança é que o algoritmo é forçado a recomeçar da melhor solução encontrada até o momento, caso após k iterações da SA não haja melhorias. Os testes computacionais foram feitos em novas instâncias geradas a partir das instâncias clássicas do PRVJT propostas por Solomon (1987). O algoritmo proposto encontrou boas soluções para instâncias com até 100 clientes.

Xu et al. (2003) desenvolveram uma heurística baseada na técnica de geração de colunas para um problema de coleta e entrega com restrições práticas, incluindo frota heterogênea, múltiplas janelas de tempo diferenciadas para coleta e entrega, incompatibilidade entre produtos e veículos, e regulamentação de horários de trabalho dos motoristas. Em instâncias geradas aleatoriamente baseando-se em características reais, o método proposto conseguiu obter soluções de boa qualidade para exemplares com até 500 requisições, em tempos computacionais aceitáveis.

Bent e Hentenryck (2006) apresentaram uma meta-heurística capaz de resolver instâncias de grande porte do PCEJT. O método é dividido em duas fases, a primeira é responsável pela diminuição de veículos utilizando SA, enquanto a segunda utiliza Busca em Vizinhança Grande (LNS, do inglês *Large Neighborhood Search*) (Shaw, 1997) para redução das distâncias percorridas. Os autores afirmam que o algoritmo de duas fases proposto é fruto da constatação de que somente a função objetivo não é suficiente para minimizar o número de veículos do problema. Muitas vezes a redução dos custos de viagem torna difícil o rearranjo para que um número menor de rotas seja encontrado em detrimento de custos mais elevados de viagem. O algoritmo em duas fases proposto foi capaz de apresentar bons resultados para instâncias com 100, 200 e 600 clientes, obtendo duas, 25 e 46 novas soluções ótimas, respectivamente.

Ropke e Pisinger (2006) projetaram uma meta-heurística chamada Busca Adaptativa em Vizinhança Grande (ALNS, do inglês *Adaptive Large Neighborhood Search*) para o PCEJT. Suas ideias foram baseadas nos trabalhos de Shaw (1997) e Bent e Hentenryck (2006). O algoritmo desenvolvido possui duas fases assim como o proposto por Bent e Hentenryck (2006). No entanto, ao invés de utilizar duas meta-heurísticas distintas, a ALNS é usada tanto na primeira fase, responsável pela redução de veículos, como na segunda fase, que minimiza as distâncias percorridas (custos de viagem). O método proposto pelos autores encontrou soluções factíveis para instâncias da literatura com mais de 500 nós, melhorando mais da metade das soluções até então encontradas.

Qu e Bard (2012) abordaram uma variante do PCEJT com transbordo, motivados pela interação com uma empresa de transporte aéreo. Uma meta-heurística Procedimento Guloso de Busca Adaptativa Aleatória (GRASP, do inglês *Greedy Randomized Adaptive Search Procedure*) (Feo e Resende, 1995), que tem como principal componente a ALNS (Ropke e Pisinger, 2006), foi proposta para resolver o problema. Também são apresentados alguns algoritmos especializa-

dos de remoção e inserção próprios para o PCEJT com transbordo. Os testes computacionais foram realizados em instâncias da literatura para o PCEJT (Li e Lim, 2001) e outras criadas pelos autores. Para as instâncias da literatura, a meta-heurística obteve resultados satisfatórios comparáveis com as melhores soluções apresentadas na literatura. Para as instâncias geradas pelos autores, o método encontrou soluções com 1% de variação para as consideradas ótimas para 88% das instâncias.

Em um trabalho posterior, Qu e Bard (2013) abordaram um problema de coleta e entrega com frota heterogênea e veículos reconfiguráveis motivado pelo transporte de idosos em tratamento médico e de reabilitação. Entende-se por veículos reconfiguráveis, aqueles que podem ter seus assentos interiores removidos, acrescentados ou redistribuídos para acomodar pacientes com diferentes necessidades de locomoção, como por exemplo, andadores e ou cadeiras de rodas. Os autores propuseram um modelo matemático contendo as características práticas relevantes e, devido à dificuldade em resolvê-lo computacionalmente por métodos exatos, desenvolveram um método heurístico de duas fases com múltiplos reinícios, que utiliza as meta-heurísticas GRASP e ALNS. De acordo com os experimentos computacionais realizados, as soluções obtidas pela abordagem proposta permitiram economias de 30 a 40% se comparadas com as rotas usadas na prática (definidas manualmente), de acordo com dados coletados durante algumas semanas de operação.

Ríos-Mercado et al. (2013) abordaram um PCEJT real oriundo de uma distribuidora de bebidas contendo várias restrições práticas como múltiplos depósitos, frota heterogênea, entrega fracionada, veículos com compartimentos, capacidade das docas e janelas de tempo nos clientes e dos veículos. As janelas de tempo dos veículos são devido a uma manutenção diária que deve ser efetuada em cada veículo da frota em um determinado intervalo de tempo. As maiores instâncias resolvidas de maneira exata utilizando o *software* de propósito geral IBM CPLEX *Optimization Studio* continham seis clientes, sete carretas e cinco produtos, isso depois de muitas horas. Em face dessa dificuldade computacional e sabendo-se do tamanho das instâncias reais, foi proposta uma meta-heurística GRASP. Em geral, a abordagem apresentada proporcionou uma redução nos custos por volta de 6% comparada com as soluções da empresa.

Como heurística construtiva para o PCEJT, Lu e Dessouky (2006) propuseram um algoritmo que difere das heurísticas construtivas convencionais, as quais consideram usualmente a redução da função objetivo em termos de custos, o que, segundo os autores, diminuem as opções de inserção que poderiam culminar em melhores soluções. Sendo assim, foi proposta uma nova forma de avaliação do nó e da posição de inserção. A função de avaliação apresentada considera o aumento do tempo de viagem e a folga da janela de tempo. Entende-se como folga da janela de tempo, a diferença entre a abertura desta e o início do serviço em determinado cliente. Outro critério que os autores incorporaram foi o de atratividade visual. Soluções que possuem uma

menor quantidade de arcos que se cruzam são preferidas pelos planejadores e tendem a ter uma menor distância (MacGregor e Ormerod, 1996; Rahimi e Dessouky, 2001). O método de inserção proposto obteve melhor desempenho do que as heurísticas construtivas convencionais em instâncias da literatura tanto para as medidas usuais e não usuais, como no caso da atratividade visual.

Ghilas et al. (2016) abordaram um PCEJT com frota heterogênea, considerando uma aplicação real envolvendo a integração de rotas de ônibus, metros e trens. Essa possibilidade de integração advém do fato que algumas rotas de veículos podem sobrepor-se a rotas de transporte público. Dessa forma, o uso de tais meios pode trazer redução de custos e de impactos ambientais, devido a menor emissão de gases poluentes. O trabalho dos autores fez uso da meta-heurística ALNS como método de solução para o problema. O algoritmo foi testado em instâncias geradas aleatoriamente, contendo 100 requisições cada e também em instâncias da literatura. Em ambos os conjuntos de instâncias, a meta-heurística conseguiu bons resultados. Neste trabalho, os autores também realizaram uma comparação dos resultados obtidos pela ALNS com os resultados encontrados por meio da implementação de um modelo em um *software* de otimização de propósito geral. Estabeleceu-se o tempo limite de 10 horas para os testes do modelo, e os resultados demonstraram que a meta-heurística conseguiu encontrar, na maioria das vezes, soluções iguais às obtidas pelo modelo em tempos muito menores. Em casos em que o modelo não conseguiu encontrar uma solução ou encontrou uma solução apenas factível, a ALNS proposta pelos autores conseguiu encontrar soluções com um *gap* de otimalidade menor, calculado utilizando-se o limitante inferior encontrado dentro do tempo estipulado de 10 horas.

Nowak et al. (2008) propuseram uma heurística para o PCEJT com coleta e entrega fracionada. A heurística cria soluções factíveis combinando busca local com características tradicionais da heurística de economias (Clarke e Wright, 1964). As cargas a serem fracionadas são selecionadas de acordo com regras pré-estabelecidas. No trabalho, os autores evidenciaram que o fracionamento da carga pode trazer redução dos custos de roteamento e até diminuir a quantidade de veículos usada. Estas evidências vão ao encontro das constatações feitas por Archetti e Speranza (2012). Os testes computacionais foram realizados usando instâncias reais e geradas aleatoriamente. Os resultados constataram que fracionar cargas ligeiramente maiores que a metade da capacidade dos veículos trouxe maior benefício. Esse benefício também foi confirmado teoricamente pelos autores. As reduções nos custos nas instâncias geradas aleatoriamente foram de 15 a 30%. No entanto, para instâncias reais, o fracionamento de carga mostrou-se pouco relevante devido a certas regras de negócio existentes como, por exemplo, custos adicionais por paradas e também devido à própria dispersão dos clientes.

Haddad et al. (2018) abordaram o PCE com coleta e entrega fracionada, tanto do ponto de vista heurístico quanto exato. Os autores propuseram um algoritmo do tipo *branch-and-price*

e uma meta-heurística baseada na Busca Local Iterada (ILS, do inglês *Iterated Local Search*). A meta-heurística faz uso de uma nova vizinhança baseada em um algoritmo de programação dinâmica, que tenta otimizar a reinserção de requisições com possível fracionamento da coleta e entrega. Em resumo, o algoritmo encontra a melhor a combinação de inserção das requisições por meio de um problema de caminho mínimo com restrição de recursos e também em um problema da mochila. A ILS proposta também seleciona vizinhanças a serem exploradas de maneira aleatória e, por fim, aplica-se um mecanismo de perturbação na melhor solução encontrada até o momento. Os testes em instâncias da literatura levaram ao uso de um único veículo. Esse fato é devido à ausência de restrições de distância, porém o método proposto pelos autores obteve soluções de boa qualidade frente a outros métodos para o mesmo tipo de problema. Para avaliar o uso de múltiplos veículos, instâncias contendo 25 a 125 requisições foram criadas. Os resultados demonstraram que o algoritmo *branch-and-price* foi capaz de lidar com instâncias de pequeno e médio porte, enquanto que a meta-heurística encontrou soluções de boa qualidade para instâncias de grande porte.

As abordagens heurísticas e meta-heurísticas mostram-se eficientes para propor soluções de boa qualidade tanto para problemas reais quanto para instâncias da literatura. No entanto, não há garantia de otimalidade das soluções encontradas por tais métodos. Os métodos exatos são capazes de certificar que quando um problema é resolvido, a solução encontrada é a melhor possível. Alguns trabalhos que fizeram uso de métodos exatos são apresentados em seguida.

Ropke et al. (2007) propuseram duas formulações extensivas, uma para o PCEJT e outra para o problema de transporte de passageiros ponto a ponto. Ambas são chamadas extensivas por possuírem número de restrições exponencial em relação ao número de requisições do problema e, portanto, requerem um método *branch-and-cut* para sua resolução. Os autores propuseram três desigualdades válidas para o PCEJT sendo elas: restrições de capacidade reforçadas (*strengthened capacity constraints*), restrições de caminhos infactíveis reforçadas (*strengthened infeasible path constraints*) e restrições do tipo garfo (*fork constrain*); e fizeram uso de outras da literatura: restrições de eliminação de sub-rotas (*sub-tour elimination constraints*), restrições de ordem generalizadas (*generalized order constraints*) e restrições de alcance (*reachability constraints*). A segunda formulação teve melhor desempenho nos testes computacionais e as desigualdades válidas mais eficazes para ambos os modelos foram as restrições de alcance e as restrições do tipo garfo. O algoritmo *branch-and-cut* utilizado foi capaz de resolver otimamente instâncias de até 96 requisições e oito veículos.

Ropke e Cordeau (2009) apresentam um algoritmo do tipo *branch-and-cut-and-price* no qual uma série de desigualdades válidas (Ropke et al., 2007) são adicionadas dinamicamente melhorando a relaxação linear do PCEJT. Os autores compararam duas abordagens para geração de colunas, a primeira é baseada no problema de caminho mínimo elementar, enquanto a segunda

é baseada na variante não-elementar deste mesmo problema. Os resultados dos testes computacionais demonstraram que a primeira estratégia resolveu 50 instâncias da literatura até a otimalidade dentro de duas horas, enquanto a segunda, resolveu 47 instâncias no mesmo tempo. As instâncias continham até 75 requisições. Outros testes foram realizados em seis instâncias com aproximadamente 500 requisições, sendo que o método proposto pelos autores resolveu três delas até a otimalidade. Outro resultado interessante que os autores obtiveram é que recorrer à geração de colunas usando o problema de caminho mínimo elementar é mais eficiente do que usar desigualdades válidas adicionais junto ao problema não-elementar.

Baldacci et al. (2011) propuseram uma abordagem exata do tipo *branch-and-cut-and-price* para o problema de coleta e entrega com janelas de tempo. Os autores estudaram o PCEJT com duas funções objetivo distintas, uma que considera somente os custos de viagem e outra que além dos custos de viagem leva em consideração o custo fixo dos veículos. O método empregado consiste em achar uma solução dual quase-ótima para a relaxação linear do problema de particionamento de conjuntos. A solução quase-ótima é calculada usando dois procedimentos que resolvem o problema dual da relaxação linear de forma aproximada, melhorando a solução de acordo com uma estratégia pré-estabelecida. Em seguida, um terceiro procedimento de geração de cortes e colunas é usado. Dessa forma, a solução obtida é usada para enumerar todas as rotas que tenham custo reduzido menor que o *gap* entre o limite superior e inferior. Por fim, o problema é resolvido por meio de *software* de otimização de propósito geral, caso tenha um número moderado de colunas, ou por um método *branch-and-cut-and-price*. Os resultados dos testes computacionais usando as instâncias propostas por Ropke e Cordeau (2009) evidenciaram que o método proposto é mais rápido que os demais disponíveis até aquele momento e foi capaz de encontrar soluções para mais 15 instâncias ainda não resolvidas até então.

Utilizando-se de um algoritmo *branch-and-price*, Bettinelli et al. (2014) abordaram o PCEJT com frota heterogênea, janelas de tempo flexíveis (nesse caso, quando uma janela de tempo é violada incide-se uma penalidade) e múltiplos depósitos. O trabalho foi motivado por um projeto real com uma pequena transportadora na região da cidade de Milão, Itália. O método proposto usa um algoritmo de programação dinâmica bidirecional. Para os testes computacionais foram consideradas janelas de tempo rígidas e flexíveis. Os resultados mostraram que o método *branch-and-price* apresentado foi capaz de solucionar e provar otimalidade para instâncias com até 75 requisições. Outra constatação foi que considerar as preferências dos clientes flexibilizando as janelas de tempo causou aumentos de 5-15% nos custos de roteamento, enquanto não observar essas preferências (características práticas) poderia levar à rejeição da solução.

Qu e Bard (2015) propuseram um método *branch-and-cut-and-price* para o PCEJT com frota heterogênea e transporte de passageiros, que nesse contexto podem ser vistos como múltiplos produtos, pois existem passageiros que ocupam o interior do veículo de diferentes maneiras (por

exemplo, ocupando assentos, espaço referente à cadeira de rodas e ou andadores). O trabalho surge motivado pela interação com uma organização de transporte de pessoas debilitadas e idosos. A formulação original para o problema tem variáveis de decisão definidas usando quatro índices. Assim, quando instâncias reais de grande porte são abordadas, sua resolução por *softwares* de otimização de propósito geral não é viável. Dessa forma, os autores usaram a decomposição de Dantzig-Wolfe, gerando um problema mestre de cobertura de conjuntos e subproblemas que são problemas de caminho mínimo com restrição de recursos. Os subproblemas são resolvidos por heurísticas e caso estas falhem em achar colunas com custos reduzidos negativos, um algoritmo de programação dinâmica com regras de dominância eficientes entre rótulos é utilizado. Cortes também são inseridos no problema mestre. Os resultados para as instâncias reais mostraram que o algoritmo é capaz de resolver problemas com até 50 clientes. Para instâncias com mais clientes, o método pode ser usado como heurística para melhorar a performance da ALNS apresentada pelos autores em outros trabalhos (Qu e Bard, 2012, 2013).

A Tabela 1 organiza os trabalhos revisados apresentando a variante do problema abordado como também o método de solução usado.

Tabela 1: Métodos de solução para o problema de coleta e entrega.

| Artigo | Problema | Método |
|-----------------------------|--------------------|----------------------------|
| (Bent e Hentenryck, 2006) | PCE/JT | Meta-heurística SA/LNS |
| (Bettinelli et al., 2014) | PCE/JT/MD/HT | Branch-and-Price |
| (Baldacci et al., 2011) | PCE/JT | Branch-and-Cut-and-Price |
| (Ghilas et al., 2016) | PCE/JT/HT | Meta-heurística ALNS |
| (Haddad et al., 2018) | PCE/SD | Meta-heurística ILS |
| (Li e Lim, 2001) | PCE/JT | Meta-heurística TS/SA |
| (Lu e Dessouky, 2006) | PCE/JT | Heurística Construtiva |
| (Qu e Bard, 2012) | PCE/JT/TR | Meta-heurística GRASP/ALNS |
| (Qu e Bard, 2013) | PCE/JT/HT/MP | Meta-heurística MALNS |
| (Qu e Bard, 2015) | PCE/JT/HT | Branch-and-Cut-and-Price |
| (Nowak et al., 2008) | PCE/SD | Heurística |
| (Ríos-Mercado et al., 2013) | PCE/JT/MD/HT/MP/SD | Meta-heurística GRASP |
| (Ropke e Pisinger, 2006) | PCE/JT/MD | Meta-heurística ALNS |
| (Ropke et al., 2007) | PCE/JT | Branch-and-Cut |
| (Ropke e Cordeau, 2009) | PCE/JT | Branch-and-Cut-and-Price |
| (Xu et al., 2003) | PCE/JT/HT/MP | Heurística |

PCE = Problema de Coleta Entrega; JT = Janelas te Tempo; MD = Múltiplos Depósitos; TR = Transbordo; HT = Frota Heterogênea; MP = Múltiplos Produtos; SD = Coleta-Entrega Fracionada.

Capítulo 3

Modelos para o PCEJT-1 e PCEJT-2

Neste capítulo, primeiramente é proposta uma formulação de dois-índices para o PCEJT-1, que incorpora restrições de frota heterogênea e múltiplos produtos. Em seguida, uma formulação de três-índices é apresentada para o PCEJT-2, que contempla as restrições de frota heterogênea, múltiplos produtos e coleta e entrega fracionada.

3.1 Modelo de dois-índices para o PCJET-1

A formulação de dois-índices proposta a seguir para o PCEJT-1 é uma extensão daquela apresentada por Furtado et al. (2017) e foi escolhida devido ao seu melhor desempenho frente a outras formulações clássicas, como já mencionado.

Devido à falta de um índice para representar os veículos, a construção da frota heterogênea dá-se pela inserção de nós artificiais de coleta e entrega (Ropke et al., 2007). Seja $m = |K|$ o número de veículos, sendo que cada veículo da frota tenha capacidade q_1, \dots, q_m e $q = \max_{1 \leq i \leq m} \{q_i\}$. Dessa forma redefine-se o número de requisições n do problema para $n + m$. Os nós artificiais de coleta são representados pelos índices $1, \dots, m$ e os artificiais de entrega por $n + 1, \dots, n + m$. Para os nós reais de coleta, sua representação é dada pelos índices $m + 1, \dots, n$ e os nós reais de entrega, são representados por $n + m + 1, \dots, 2n$. A Figura 2 apresenta um grafo contendo nós reais para um problema com duas requisições ($n = 2$), e um segundo grafo após a inserção de uma requisição artificial ($m = 1$). Sendo assim, o problema passa a ter três requisições, onde os nós reais de coleta são representados pelos nós 2 e 3 e os nós reais de entrega por 5 e 6. Enquanto o nó artificial de coleta é representado por 1 e o nó artificial de entrega por 4.

Para o caso de múltiplos produtos, define-se primeiramente um índice $l = 1, \dots, L$, representando os produtos do problema e uma variável inteira positiva y_{il} que indica a carga do produto l após deixar o nó i . O parâmetro d_{il} representa a demanda de um produto l por um nó i de coleta; em contrapartida, tem-se um parâmetro $d_{n+i,l} = -d_{il}$, que define a demanda de um produto l para um nó de entrega $n + i$. Cada produto l tem um parâmetro associado u_l que é o fator multiplicador para conversão da unidade de medida do produto para a capacidade dos veículos. Por exemplo, se uma unidade do produto l pesa 1kg e a capacidade dos veículos da frota é dada em toneladas, então o parâmetro u_l tem valor $\frac{1}{1000} = 0,001$. Também é necessário a criação de um produto artificial l_a , tal que $d_{i,l_a} = q - q_i$ e $d_{n+i,l_a} = -d_{i,l_a}$, para todo nó artificial

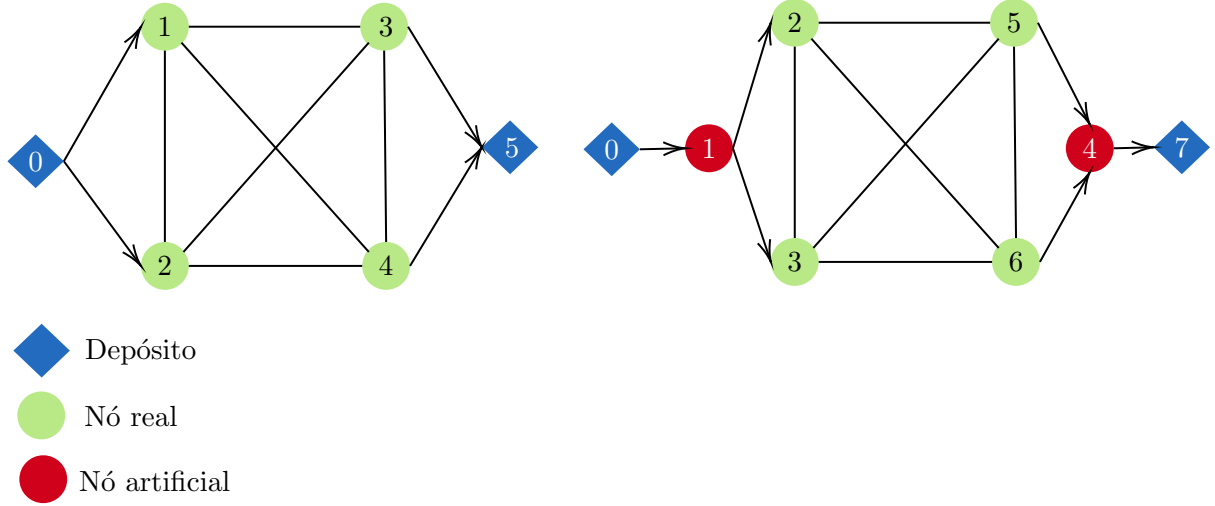


Figura 2: Grafo com nós reais e após a inserção dos nós artificiais.

$i = 1, \dots, m$. A demanda para todos os produtos reais dos respectivos nós artificiais deve ser 0, enquanto que, para todos os nós reais a demanda pelo produto artificial também deve ser 0.

Devido à inserção dos nós artificiais ao problema, é necessário impor novos custos e tempos de viagem a determinados arcos do problema. Dessa forma, deve-se garantir que todos os arcos que deixam o depósito inicial e vão para os nós de coleta artificiais tenham custos e tempos de viagem iguais a zero. O mesmo deve ser feito para os custos e tempos de viagem para todos os arcos que ligam os nós de entrega artificiais ao depósito final. Arcos entre os nós artificiais, também devem ter custos e tempos de viagem iguais a zero. Os arcos entre os nós de coleta artificiais e os reais j devem ter custos c_{0j} e tempos de viagem t_{0j} . Enquanto os custos e tempos de viagem dos nós de entrega reais $n + j$ para os de entrega artificiais devem corresponder a $c_{n+j,2n+1}$ e $t_{n+j,2n+1}$ (Ropke et al., 2007). A Figura 3 ilustra os novos custos e tempos de viagens assumidos. Nesta figura, k e $n + k$ são os nós artificiais de coleta e entrega, enquanto os demais são nós reais. Por simplicidade, assume-se somente uma requisição artificial e duas requisições reais, no entanto, o problema pode conter várias requisições artificiais e reais. A formulação de dois-índices para o PCEJT-1 é apresentada a seguir:

$$\min \sum_{i=0}^{2n+1} \sum_{j=0}^{2n+1} c_{ij} x_{ij} \quad (3.1a)$$

$$\text{s.a} \quad \sum_{\substack{i=0 \\ i \neq j}}^{2n} x_{ij} = 1, \quad j = 1, \dots, 2n + 1, \quad (3.1b)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{2n+1} x_{ij} = 1, \quad i = 0, \dots, 2n, \quad (3.1c)$$

$$w_j \geq w_i + t_{ij} - M(1 - x_{ij}), \quad i, j = 0, \dots, 2n + 1, \quad (3.1d)$$

$$w_i^a \leq w_i \leq w_i^b, \quad i = 0, \dots, 2n + 1, \quad (3.1e)$$

$$y_{jl} \geq y_{il} + d_{jl} - M(1 - x_{ij}), \quad i, j = 0, \dots, 2n + 1; l = 1, \dots, L, \quad (3.1f)$$

$$\sum_{l=1}^L y_{il} \cdot u_l \leq q, \quad i = 0, \dots, 2n + 1, \quad (3.1g)$$

$$w_{n+i} \geq w_i + s_i + t_{i,i+n}, \quad i = 1, \dots, n, \quad (3.1h)$$

$$v_{n+i} = v_i, \quad i = 1, \dots, n, \quad (3.1i)$$

$$v_j \geq j \cdot x_{0j}, \quad j = 1, \dots, 2n, \quad (3.1j)$$

$$v_j \leq j \cdot x_{0j} - n(x_{0j} - 1), \quad j = 1 \dots, 2n, \quad (3.1k)$$

$$v_j \geq v_i + n(x_{ij} - 1), \quad i, j = 1 \dots, 2n, \quad (3.1l)$$

$$v_j \leq v_i + n(1 - x_{ij}), \quad i, j = 1 \dots, 2n, \quad (3.1m)$$

$$x_{0i} = 0, \quad i = m + 1, \dots, 2n + 1, \quad (3.1n)$$

$$x_{ij} = 0, \quad i = 1, \dots, m; j = 0, \dots, m, n + m + 1, \dots, 2n, \quad (3.1o)$$

$$x_{ij} = 0, \quad i = m + 1, \dots, n + m; j = n + 1, \dots, n + m, \quad (3.1p)$$

$$x_{i,2n+1} = 0, \quad i = 0, \dots, n, n + m + 1, \dots, 2n. \quad (3.1q)$$

$$x_{ij} \in \{0, 1\}, \quad i, j = 0, \dots, 2n + 1. \quad (3.1r)$$

As restrições de (3.1a) a (3.1m) têm a mesma interpretação referente à formulação (2.7). As restrições (3.1n) garantem que o depósito inicial só tenha ligações com os nós artificiais de coleta. As restrições (3.1o) determinam que os nós de coleta artificiais só tenha ligações com os nós reais de coleta e com os artificiais de entrega. Restrições (3.1p) garantem que os nós artificiais de entrega só possam ser alcançados por meio dos nós de entrega reais e pelos de coleta artificiais. Restrições do tipo (3.1q) garantem que o depósito final só tenha ligações com os nós de entrega artificiais. Por fim, as restrições (3.1r) determinam o domínio das variáveis de decisão.

3.2 Modelo de três-índices para o PCEJT-2

A formulação clássica de três-índices (2.6), apresentada na Seção 2.2, pode ser estendida para considerar frota heterogênea, múltiplos produtos e coleta e entrega fracionada (Golden et al., 2008; Archetti e Speranza, 2012; Qu e Bard, 2013). As definições apresentadas na Seção 3.1 sobre múltiplos produtos também são válidas para o PCEJT-2. Dessa forma, é necessário adicionar ao modelo uma variável inteira positiva y_{ikl} que indica a carga do produto l no veículo k após deixar o nó i . No caso da coleta e entrega fracionada temos a inserção de uma nova variável de decisão z_{ikl} que representa a quantidade do produto l coletada ou entregue por um veículo k ao nó i . Diferentemente dos modelos anteriores, assume-se também que a demanda de um nó de entrega não seja negativa e dessa forma temos que $d_{n+i,l} = d_{i,l}$ para todo $i = 1, \dots, 2n$ e $l = 1, \dots, L$. A seguir, a formulação de três-índices proposta para o PCEJT-2 é apresentada.

$$\min \sum_{k=1}^K \sum_{i=0}^{2n+1} \sum_{j=0}^{2n+1} c_{ij} x_{ijk} \quad (3.2a)$$

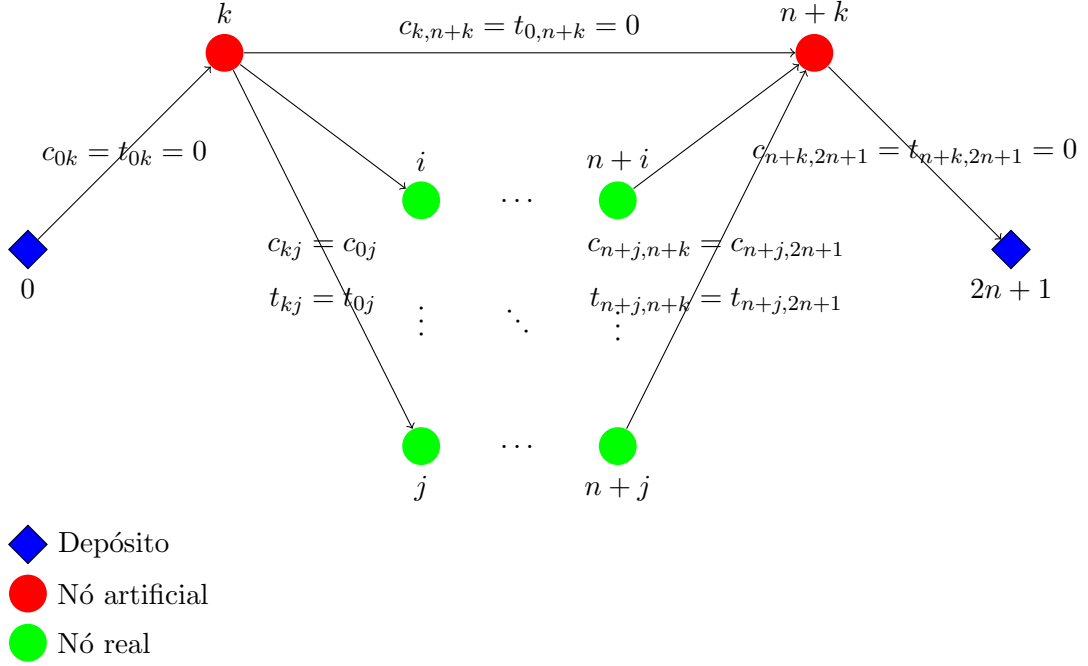


Figura 3: Nova estrutura do grafo contendo nós artificiais.

$$\text{s.a} \quad \sum_{k=1}^K \sum_{\substack{j=1 \\ j \neq i}}^{2n+1} x_{ijk} \geq 1, \quad i = 1, \dots, n, \quad (3.2b)$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{2n} x_{ijk} - \sum_{\substack{j=1 \\ n+i \neq j}}^{2n+1} x_{n+i,j,k} = 0, \quad k = 1, \dots, K; i = 1, \dots, n, \quad (3.2c)$$

$$\sum_{j=1}^{2n+1} x_{0jk} \leq 1, \quad k = 1, \dots, K, \quad (3.2d)$$

$$\sum_{j=1}^n x_{0jk} - \sum_{i=n+1}^{2n} x_{i,2n+1,k} = 0, \quad k = 1, \dots, K, \quad (3.2e)$$

$$\sum_{\substack{j=0 \\ j \neq 1}}^{2n} x_{jik} - \sum_{\substack{j=1 \\ i \neq j}}^{2n+1} x_{ijk} = 0, \quad k = 1, \dots, K; i, j = 1, \dots, 2n, \quad (3.2f)$$

$$w_{jk} \geq w_{ik} + (s_i + t_{ij})x_{ijk} - M(1 - x_{ijk}), \quad k = 1, \dots, K; i, j = 0, \dots, 2n+1, \quad (3.2g)$$

$$w_{ik} + s_i + t_{i,n+i} \leq w_{n+i,k}, \quad k = 1, \dots, K; i = 1, \dots, 2n, \quad (3.2h)$$

$$w_i^a \leq w_{ik} \leq w_i^b, \quad k = 1, \dots, K; i = 1, \dots, 2n, \quad (3.2i)$$

$$y_{jkl} \geq y_{ikl} + z_{jkl} - M(1 - x_{ijk}), \quad k = 1, \dots, K; i = 0, \dots, 2n; \quad (3.2j)$$

$$j = 1, \dots, n; l = 1, \dots, L,$$

$$y_{jkl} \geq y_{ikl} - z_{jkl} - M(1 - x_{ijk}), \quad k = 1, \dots, K; i = 0, \dots, 2n; \quad (3.2k)$$

$$j = n+1, \dots, 2n; l = 1, \dots, L,$$

$$\sum_{l=1}^L y_{ikl} \cdot u_l \leq q_k, \quad k = 1, \dots, K; i = 0, \dots, 2n+1; \quad (3.2l)$$

$$l = 1, \dots, L,$$

$$z_{ikl} \leq d_{il} \sum_{\substack{j=0 \\ i \neq j}}^{2n+1} x_{jik}, \quad \begin{array}{l} k = 1, \dots, K; i = 0, \dots, 2n + 1; \\ l = 1, \dots, L, \end{array} \quad (3.2m)$$

$$\sum_{k=1}^K z_{ikl} = d_{il}, \quad i = 0, \dots, 2n + 1; l = 1, \dots, L, \quad (3.2n)$$

$$x_{ijk} \in \{0, 1\}, \quad k = 1, \dots, K; i, j = 0, \dots, 2n + 1. \quad (3.2o)$$

A função objetivo (3.2a) consiste em minimizar os custos de roteamento. As restrições (3.2b) asseguram que os clientes sejam visitados ao menos uma vez. As restrições (3.2d) garantem que os veículos usados deixem o depósito uma única vez. O balanceamento de carga dos veículos após a visita aos clientes é feito pelas restrições (3.2j) e (3.2k). A garantia de que a capacidade dos veículos não será extrapolada é assegurada pelas restrições (3.2l). As restrições (3.2m) garantem que a quantidade de um produto coletado ou entregue por um veículo ao se visitar um cliente seja menor ou igual a sua demanda pelo respectivo produto. Restrições (3.2n) asseguram que a soma das quantidades coletadas ou entregues de um determinado produto pelos veículos a um cliente seja igual à demanda deste pelo respectivo produto. As restrições (3.2o) asseguram o domínio das variáveis de decisão. As demais restrições têm a mesma interpretação daquelas apresentadas no modelo (2.6).

Capítulo 4

Meta-Heurística para o PCEJT-1 e PCEJT-2

Neste capítulo, propõe-se uma meta-heurística do tipo Busca Adaptativa em Vizinhança Grande (ALNS), que consegue lidar com restrições práticas referentes à frota heterogênea, múltiplos produtos e coleta e entrega fracionada. O método foi baseado nos trabalhos de Ropke e Pisinger (2006), Bent e Hentenryck (2006), Qu e Bard (2013) e Ghilas et al. (2016), sendo que Ropke e Pisinger (2006) propuseram a meta-heurística ALNS inicialmente utilizando ideias do método apresentado por Bent e Hentenryck (2006). Os dois primeiros trabalhos abordaram o PCEJT clássico, enquanto que os trabalhos de Qu e Bard (2013) e Ghilas et al. (2016) apresentaram uma implementação da ALNS com adaptações para frota-heterogênea e múltiplos produtos. A ALNS também se mostrou promissora para a resolução de outros tipos de problemas de roteamento de veículos (Pisinger e Ropke, 2007, 2010).

Para melhor compreensão da meta-heurística ALNS, pode-se dizer que ela consiste basicamente em uma Busca em Vizinhança Grande (LNS) que, similarmente à metodologia de destruição e reconstrução proposta por Schrimpf et al. (2000), realoca até 40% das requisições de uma solução a cada iteração. As diferenças entre uma ALNS e uma LNS clássica é que ao invés de um operador de destruição e reconstrução, a ALNS emprega vários operadores e a sua seleção é baseada na coleta e uso de informações sobre o desempenho obtido por estes ao longo da busca, o que dá origem ao termo *adaptativo*. Além disso, é comum que as implementações da LNS sejam descendentes, ou seja, aceita-se sempre uma solução que tenha custo menor que a atual, enquanto a implementação da ALNS usa um critério de aceite da SA.

De modo geral, a ALNS parte de uma solução inicial e por meio dos operadores de destruição e reconstrução tenta-se encontrar uma solução vizinha que seja melhor que a anterior. Avalia-se que uma solução é melhor que uma anterior de acordo com um critério pré-estabelecido e a busca continua até que um critério de parada seja atingido. O Algoritmo 1 apresenta o funcionamento geral da ALNS. Inicialmente, o algoritmo recebe uma solução inicial S , gerada por uma heurística construtiva, o número máximo de iterações $maxIterations$ e o tempo máximo permitido para execução do procedimento de busca $timeToSpend$. Em seguida, a variável contadora que controla o número de iterações para atualização dos pesos das heurísticas $contIt$ é iniciada (linha 1). Antes do início do processo de busca, estabelece-se também que a solução inicial S passa ser a melhor solução S_{best} (linha 2). Os pesos das heurísticas de remoção e inserção são atualizados a cada 100 iterações (linhas 5 à 8). Primeiramente, escolhe-se um

número μ de requisições para remoção (linha 9). Em seguida, uma heurística de remoção e inserção é selecionada de acordo com seus pesos (linha 10). Posteriormente, a heurística de remoção remove uma quantidade pré-estabelecida de requisições da solução incumbente (linha 11) que são reinseridas na solução pelas heurísticas de inserção (linha 11). Em seguida avalia-se se a solução incumbente S_* tem valor menor que a melhor solução S_{best} , em caso afirmativo ela passa a ser a melhor solução (linhas 13 à 15). De qualquer forma, todas as soluções passam pelo critério de aceite da SA que estabelece se a nova solução passa ser a solução incumbente explorada (linhas 16 à 18). Por fim, a variável *countIt* tem seu valor incrementado (linha 19). O algoritmo é executado até que atingir um número máximo de iterações ou esgotar o tempo concedido para a busca (linha 20).

Algoritmo 1: ALNS

Entrada: $S, maxIterations \in \mathbb{N}, timeToSpend \in \mathbb{N}$;
Saída: S_{best} ;

- 1 $countIt = 0$;
- 2 $S_{best} = S$;
- 3 **repita**
- 4 $S_* = S$;
- 5 **se** $countIt = 100$ **então**
- 6 atualize os pesos das heurísticas;
- 7 $countIt = 0$;
- 8 **fim**
- 9 escolha um número μ entre [10%; 40%] do número de requisições do problema ;
- 10 selecione uma heurística de remoção h_r e inserção h_i ;
- 11 remova μ requisições de S_* utilizando h_r e as insira em U ;
- 12 insira as requisições contidas em U em S_* utilizando h_i ;
- 13 **se** $S_* \leq S_{best}$ **então**
- 14 $S_{best} = S_*$;
- 15 **fim**
- 16 **se** $aceito(S_*, S)$ **então**
- 17 $S = S_*$;
- 18 **fim**
- 19 $countIt = countIt + 1$;
- 20 **até** $maxIterations \geq 0$ E $timeToSpend > 0$;

Tanto Ropke e Pisinger (2006) como Bent e Hentenryck (2006) utilizaram um algoritmo com duas fases. A primeira fase reduz o número de veículos, enquanto a segunda minimiza os custos

de roteamento. Ambos os trabalhos abordam problemas com frota homogênea e de acordo com Ropke e Pisinger (2006) essa abordagem não consegue lidar com frota heterogênea. Nos trabalhos de Qu e Bard (2012, 2013), os autores propuseram uma heurística específica para redução de veículos, devido à natureza heterogênea da frota encontrada nos problemas abordados, fazendo que o algoritmo tenha somente uma fase. Da mesma forma que estes autores, implementou-se neste trabalho uma heurística específica para redução de veículos.

A ALNS proposta faz uso de heurísticas de remoção para destruir parte da solução ao passo que a cada iteração remove-se μ requisições da atual solução S , movendo-as para um conjunto U denominado *banco de requisições*. Esse conjunto é de extrema utilidade por permitir que a ALNS visite soluções infactíveis criando diversificação para melhora das soluções. Por meio do procedimento de reconstrução, as heurísticas de inserção reinsere todas as requisições que estão em U . As heurísticas de remoção foram baseadas nos trabalhos de Ropke e Pisinger (2006), Qu e Bard (2012, 2013) e Ghilas et al. (2016). Exceto as heurísticas responsáveis pelo fracionamento de coleta e entrega, as heurísticas de inserção foram implementadas de acordo com Ropke e Pisinger (2006), que são heurísticas simples e rápidas.

As inserções são avaliadas pelo custo, $c(S)$, da função objetivo representada por:

$$c(S) = w_1 \sum_{r \in S} \sum_{(i,j) \in r} t_{ij} + w_2 \sum_{r \in S} I_r + w_3 \sum_{i \in U} H_i, \quad (4.1)$$

sendo r uma rota factível pertencente a S ; t_{ij} é o tempo de deslocamento do nó i para j conforme definido anteriormente; I_r uma variável que assume valor 0 caso a rota esteja vazia (não possua nenhum nó de coleta e entrega, apenas o depósito inicial e final), ou 1 caso contrário; H_i uma variável que assume valor 0 caso a requisição i esteja na solução S , ou 1 caso contrário. Os termos da função objetivo da ALNS são ponderados por pesos w_1, w_2 e w_3 , cuja ordem de grandeza é $w_1 \ll w_2 \ll w_3$. Ao atribuir pesos aos termos da função objetivo, espera-se que os mesmos garantam a factibilidade das soluções encontradas pela ALNS (Ropke e Pisinger, 2006; Qu e Bard, 2012, 2013). Dessa forma, a função objetivo consiste em minimizar primeiramente as requisições não roteadas do problema, em seguida o número de veículos e por fim os tempos/distâncias percorridas. O critério de aceite, o mecanismo adaptativo, as heurísticas de remoção e inserção são descritos a seguir.

4.1 Critério de aceite

De acordo com Ropke e Pisinger (2006), implementações usuais da LNS que aceitam somente soluções melhores que a atual são propensas a ficar presas em ótimos locais. O critério de aceite da SA permite que soluções piores que a atual possam vir a ser aceitas de acordo com a probabilidade $e^{-\frac{S_* - S}{T}}$, onde T é a temperatura inicial.

A temperatura inicial é calculada por:

$$T = \frac{c(S) * w}{n \log 2}. \quad (4.2)$$

A expressão é nada mais que estabelecer a temperatura inicial para que uma solução $w\%$ pior que uma solução inicial S , seja aceita com uma probabilidade de 0,5. Dividindo-se posteriormente o resultado pelo número de requisições do problema, n , calcula-se a temperatura inicial. Realizar a divisão pelo número de requisições do problema permite que a temperatura inicial seja mais adequada a problemas de vários tamanhos (Ropke e Pisinger, 2006; Pisinger e Ropke, 2007). A cada iteração, a temperatura é resfriada conforme um fator α de maneira que a nova temperatura seja $T = \alpha.T$ e $T \geq 0$.

4.2 Mecanismo adaptativo

O processo de busca da ALNS é feito destruindo e reconstruindo partes da solução. Para isso, a cada iteração, heurísticas de remoção e inserção são usadas. Recorda-se que uma das diferenças da ALNS para uma LNS é o número de operadores de destruição e reconstrução. Seja k o número de heurísticas implementadas na ALNS. A cada heurística i temos um peso associado W_i , $i \in \{1, 2, \dots, k\}$.

A meta-heurística seleciona a cada iteração uma heurística de remoção e uma de inserção, baseando-se na performance e peso das heurísticas em iterações passadas. Primeiramente, divide-se a busca em *segmentos*, de modo que um segmento represente 100 iterações da meta-heurística. O próximo passo é coletar as informações necessárias para avaliar o desempenho das mesmas. As informações são: o número de vezes que uma heurística i foi selecionada no segmento passado, representado por θ_i ; e a pontuação π_i da heurística no segmento anterior definida conforme discutido abaixo. De posse dessas informações, os pesos das heurísticas são atualizados de acordo com:

$$W_{i,j+1} = (1 - r) * W_{ij} + r \frac{\pi_i}{\theta_i}. \quad (4.3)$$

Na expressão (4.3), W_{ij} é o peso da heurística i no segmento j e r é um fator de reação: se $r = 0$ a pontuação da heurística no segmento passado é desconsiderada e somente o peso inicial é usado para avaliar qual heurística será selecionada, enquanto para $r = 1$, o peso da heurística em segmentos passados é negligenciado e somente a atual pontuação é usada. Pontua-se cada heurística por σ_1, σ_2 e σ_3 para as seguintes situações:

- σ_1 : caso as operações de remoção e inserção encontrarem uma solução melhor que a global;
- σ_2 : caso as operações de remoção e inserção encontrarem uma solução que foi aceita e o seu custo é melhor que a solução atual; e

- σ_3 : caso as operações de remoção e inserção encontrarem uma solução que foi aceita, no entanto seu custo é pior que o da solução atual.

Todas as soluções aceitas devem obrigatoriamente ter todas as requisições roteadas, ou seja, o banco de requisições U deve estar vazio.

As heurísticas são selecionadas de acordo com o *princípio da roleta*, que consistem em selecionar um indivíduo de acordo com certa probabilidade. Sendo assim, sorteia-se um número aleatório dentro do intervalo $[0,1]$ e se verifica, de acordo com a probabilidade acumulada, em qual faixa o número sorteado se encontra (Ropke e Pisinger, 2006; McCaffrey, 2017). A heurística de remoção é selecionada de maneira independente da heurística de inserção e vice-versa. Inicialmente, todas as heurísticas têm peso igual a um. Uma heurística j é selecionada com probabilidade:

$$\frac{W_j}{\sum_{i=1}^k W_i}. \quad (4.4)$$

4.3 Heurísticas de remoção

A meta-heurística faz uso de oito heurísticas de remoção, cada uma com o propósito de remover μ requisições da solução. Dessa maneira, elas exploram oito diferentes vizinhanças. As heurísticas foram baseadas nos trabalhos de Ropke e Pisinger (2006), Qu e Bard (2012, 2013) e Ghilas et al. (2016). Quando uma requisição i é removida, implica na retirada do nó de coleta i e do nó de entrega $n + i$ da solução. O valor de μ é calculado selecionando-se um número aleatório dentro do intervalo $[0,1n; 0,4n]$. Todas as requisições removidas pelas heurísticas são automaticamente inseridas no banco de requisições U (Ropke e Pisinger, 2006).

4.3.1 *Shaw Removal* e variantes

De acordo com um fator de semelhança R_{ij} entre duas requisições, i e j , a *Shaw removal* remove requisições que são parecidas umas com as outras. Parte-se do pressuposto que é mais fácil reinserir as requisições em posições diferentes caso elas sejam similares. É importante dizer que, duas requisições distintas, i e j , não necessariamente precisam pertencer a rotas diferentes.

Primeiramente uma requisição i é escolhida ao acaso e calcula-se o fator de semelhança para as outras requisições j pertencentes à S por (4.5). Essa medida considera tempos de viagem t_{ij} , tempos de início de serviço w_i e demanda dos produtos $d_{i,l}$. Os pesos φ , χ e ψ são normalizados, ou seja, então entre $[0,1]$. Ordenam-se essas requisições em uma lista OL de maneira crescente (quanto menor R_{ij} mais parecida são as requisições). Escolhe-se um número aleatório $p \in [0, 1]$. A escolha da requisição para remoção é feita por $OL[p^\delta \cdot |OL|]$, onde $\delta \in \mathbb{R}_+$ é um parâmetro usado para criar aleatoriedade no processo de remoção. A expressão $p^\delta \cdot |OL|$ é usada para calcular a posição na lista onde situa-se a requisição a ser removida. Nas iterações subsequentes

a heurística remove requisições semelhantes com as já removidas selecionando ao acaso uma requisição dentro de U .

$$R_{ij} = \varphi(t_{ij} + t_{n+i,n+j}) + \chi(|w_i - w_j| + |w_{n+i} - w_{n+j}|) + \psi \left| \sum_{l=1}^L d_{i,l} - \sum_{l=1}^L d_{j,l} \right|. \quad (4.5)$$

Além da *Shaw Removal*, são usadas outras três heurísticas de remoção denominadas *Time Related Removal*, *Distance Related Removal* e *Demand Related Removal* que são casos especiais da *Shaw Removal* com pesos $\chi = 1$, $\varphi = 1$ e $\psi = 1$ respectivamente enquanto os outros pesos são 0.

4.3.2 *Late Arrival Removal*

A heurística de remoção *Late Arrival Removal* considera remover requisições que tenham a maior diferença entre o início do serviço e abertura da janela de tempo, tanto para nós de coleta quanto de entrega. Primeiramente, as requisições são ordenadas de forma decrescente com relação a essa diferença, LA_i , que é calculada pela expressão (4.6). Utiliza-se em seguida o mesmo mecanismo de escolha da *Shaw Removal* para a próxima requisição que será removida da lista ordenada.

$$LA_i = |w_i - w_i^a| + |w_{n+i} - w_{n+i}^a|. \quad (4.6)$$

4.3.3 *Random Removal*

A heurística *Random Removal* é a mais simples de todas as heurísticas de remoção. Ela seleciona ao acaso as requisições e as remove da solução.

4.3.4 *Worst Removal*

Seja $c(S, i)$ o custo da solução onde a requisição i é removida do problema. Seja $\Delta c(S, i) = c(S) - c(S, i)$, a variação do custo da solução caso i seja retirada. A heurística *Worst Removal* remove a cada iteração a requisição com o maior valor de $\Delta c(S, i)$. O objetivo principal desta heurística é remover requisições que devido a seus custos altíssimos parecem que estão em uma posição errada, arruinando a estrutura da solução atual. As requisições são colocadas em uma lista *OL*, ordenada de maneira decrescente em função de $\Delta c(S, i)$. A escolha da próxima requisição que deve ser removida é feita da mesma forma usada na *Shaw Removal*, usando um número aleatório p e um parâmetro δ , assim um grau de aleatoriedade é adicionado evitando que uma mesma requisição venha a ser repetidamente removida. O procedimento da heurística é apresentado no Algoritmo 2.

4.3.5 *Random Route Removal*

Como mencionado anteriormente, o algoritmo em duas fases proposto por Ropke e Pisinger (2006) não lida com frota heterogênea. A heurística apresentada neste capítulo é baseada nos

Algoritmo 2: Worst Removal

Entrada: $S \in \text{Solucoes}$, $\mu \in \mathbb{N}$;

1 repita

2 | lista ordenada : $OL =$ todas requisições i roteadas, ordenadas segundo $\Delta c(S, i)$;

3 | escolha um número aleatório $p \in [0,1]$;

4 | requisição : $\mathbf{r} = OL[p^\delta \cdot |OL|]$;

5 | remova \mathbf{r} de S ;

6 | $\mu = \mu - 1$;

7 até $\mu = 0$;

trabalhos de Qu e Bard (2012, 2013) que propõem implementações da ALNS abordando frota heterogênea. O algoritmo da *Random Route Removal* seleciona aleatoriamente uma rota r pertencente à solução atual do problema e remove todos os nós de r , deixando somente os nós 0 e $2n + 1$ que são os depósitos inicial e final respectivamente. A rota é deixada vazia para garantir factibilidade caso não seja possível reinserir os nós em outras rotas. Cada vez que a *Random Route Removal* é selecionada aumenta-se a chance de encontrar soluções com um menor número de veículos. Lembrando que cada heurística de remoção deve remover μ requisições da solução. Caso ainda haja requisições para completar o número μ após o término das iterações da *Random route removal*, a *Shaw removal* é usada para se chegar ao valor desejado. O Algoritmo 3 apresenta os passos da heurística.

Algoritmo 3: Random route removal

Entrada: $S \in \text{Solucoes}$, $\mu \in \mathbb{N}$;

1 rota : $\mathbf{r} =$ uma rota selecionada aleatoriamente de S que não esteja vazia;

2 vetor de requisições : $V = \{\text{todas as requisições contidas em } \mathbf{r}\}$;

3 remova as requisições em V de S e consecutivamente de \mathbf{r} ;

4 $\mu = \mu - |V|$;

5 se $\mu > 0$ **então**

6 | $\text{Shaw Removal}(S, \mu)$;

7 fim

4.4 Heurísticas de inserção

Ao todo, foram implementadas seis heurísticas de inserção, sendo que uma delas é responsável por incorporar o fracionamento da coleta e entrega. Na verdade, cada heurística é dividida em duas, uma analisa os custos de inserção simplesmente pela função objetivo, enquanto a outra adiciona um ruído, ξ , no cálculo do custo da solução. Esse ruído é necessário para criar

diversificação no processo de busca por melhores soluções e, com o critério de aceite de SA, evita que o algoritmo fique preso em ótimos locais. O custo modificado pelo ruído é calculado por $c(S)^* = \max\{0, c(S) + \xi\}$, sendo que ξ é escolhido aleatoriamente dentro do intervalo $[-\max R, \max R]$, onde $\max R = \eta \cdot \max_{i,j \in N} \{t_{ij}\}$, N é o conjunto de todos os nós do problema e η é um parâmetro que controla o valor do ruído. Como o tempo de viagem é usado na função objetivo, empregar esta medida no cálculo do nível do ruído o torna de certa forma proporcional à função objetivo.

4.4.1 *Regret heuristics*

Para as heurística de arrependimento, *regret heuristics*, avalia-se o custo de inserção de uma requisição i na sua melhor posição possível na melhor rota, $c(S, i)_{best}$, e na sua k -ésima melhor posição e rota, $c(S, i)_k$. Entende-se por, a melhor posição possível na melhor rota, a inserção que leva ao menor custo em termos da função objetivo $c(S)$. Este tipo de heurística busca de certa maneira “olhar para o futuro”, na tentativa de avaliar se caso a requisição i não for inserida imediatamente, quão custoso (maior o arrependimento) será inseri-la posteriormente. Dessa forma, define-se uma classe de heurísticas de arrependimento, *regret* - k , e a cada iteração a heurística busca inserir a requisição i que tenha o maior arrependimento:

$$\max_{i \in U} \left\{ \sum_{j=1}^k c(S, i)_j - c(S, i)_{best} \right\}. \quad (4.7)$$

Sempre as requisições são inseridas na sua melhor posição e as heurísticas realizam no máximo n iterações a cada vez que são escolhidas. As implementações variam $k = \{1, 2, 3, 4, m\}$, sendo m o número de rotas da solução S . Ressalta-se que a heurística com $k = 1$ é um algoritmo guloso, em que insere-se a requisição mais barata dentre todas, enquanto que para $k = m$ avaliam-se todas as rotas e posições para inserção. Caso alguma requisição não possa ser inserida em $m - k + 1$ rotas, então a requisição que possa ser inserida no menor número de rotas é inserida.

4.4.2 *Split heuristics*

As heurísticas usadas para realizar o fracionamento da coleta e entrega de uma requisição assemelham-se à heurística de arrependimento com $k = 1$. A mudança é que as requisições são inseridas tendo sua demanda dividida entre duas rotas. No caso de múltiplos produtos, o número de produtos que terão sua demanda fracionada é sorteado dentro do intervalo de $[1, |L|]$. Para todo produto que será fracionado, uma quantidade $d_{i,l}^1$ de sua demanda é sorteada dentro do intervalo $[1, d_{i,l} - 1]$. Dessa forma, uma rota é responsável por coletar e entregar $d_{i,l}^1 \in [1, d_{i,l} - 1]$, enquanto uma segunda fica encarregada de coletar e entregar $d_{i,l}^2 = d_{i,l} - d_{i,l}^1$, para todo produto fracionado.

4.5 Heurística construtiva

A solução inicial usada na implementação da meta-heurística proposta neste trabalho é gerada da seguinte maneira. Inicialmente todas as requisições são colocadas em U . Os veículos são ordenados em uma lista em ordem decrescente em função da sua capacidade. O primeiro veículo da lista é atribuído a uma rota e é retirado da lista, evitando que o mesmo seja usado posteriormente. Uma heurística de inserção que seleciona aleatoriamente uma requisição e a insere na melhor posição possível é executada n vezes, ou seja, até que o banco de requisições U esteja vazio. A função objetivo usada por esta heurística sempre tem adição de ruído e todas as rotas são avaliadas. Caso o problema permita coleta e entrega fracionada, uma *split heuristic* é chamada após a execução da heurística de inserção, se ainda houver requisições não roteadas, realizando o fracionamento das mesmas. O algoritmo do procedimento de construção da solução inicial é executado até que todas as requisições tenham sido roteadas ou até que a frota de veículos se esgote. Neste último caso, a ALNS inicia com requisições dentro de U .

Capítulo 5

Resultados

Neste capítulo, são apresentados a calibração dos parâmetros e os resultados de testes computacionais com os modelos e meta-heurística propostos nos Capítulos 3 e 4. Os testes foram realizados em um computador com processador Intel Core i7-7500U, 2.70GHz, com 16GB de memória RAM e sistema operacional Windows 10. O solver utilizado foi o IBM Cplex 12.7.0, por meio da biblioteca *Concert*. A codificação dos modelos e da meta-heurística foi feita em C++. Segundo os demais trabalhos da literatura, a saída dos veículos do depósito é penalizada em 10^4 , sendo este também o valor do parâmetro w_2 da ALNS. O valor dos outros parâmetros são $w_1 = 1$ e $w_3 = 10^6$. Cabe lembrar que os parâmetros w_1 , w_2 e w_3 são os pesos para ponderação dos termos da função objetivo da ALNS que, respectivamente, consideram as distâncias/tempos percorridos, o número de veículos e as requisições não roteadas do problema.

5.1 Descrição das instâncias

Para os testes computacionais dos modelos e da meta-heurística proposta foram utilizadas as instâncias apresentadas por Ropke e Cordeau (2009). Nestas instâncias, considera-se uma área de dimensões $[0, 50] \times [0, 50]$, onde cada localização dos nós de coleta e entrega é disposta aleatoriamente seguindo uma distribuição uniforme. Os valores de c_{ij} e t_{ij} foram gerados por distâncias Euclidianas e não dependem do tipo de veículo. No que diz respeito à demanda d_i de cada cliente, foram selecionadas aleatoriamente do intervalo $[5, q]$, onde q é um parâmetro que depende de cada classe conforme definido mais a frente. O horizonte de planejamento tem duração total $T = 600$, o intervalo de abertura e fechamento das janelas de tempo foi gerado escolhendo w_i^a aleatoriamente dentro do intervalo $[0, T - t_{i,n+i}]$, posteriormente $w_i^b = w_i^a + w$, sendo w um parâmetro que determina a duração da janela de tempo. Para os nós de entrega $w_{n+i}^a = w_i^a + t_{i,n+i}$ e $w_{n+i}^b = w_{n+i}^a + w$. Há quatro classes de instâncias, AA, BB, CC e DD, os valores de q e w para cada uma delas é dado pela Tabela 2.

Para todas as classes existem originalmente 10 instâncias. No entanto, utilizou-se o mesmo procedimento descrito por Furtado et al. (2017) e dessa maneira novas instâncias contendo n requisições foram criadas, com $n = 5, 10, 15, 20, 25$. As primeiras n requisições são selecionadas de cada instância com 30 requisições de cada classe. Assim há 15 instâncias para cada classe, por exemplo: AA5, AA10, ..., AA75, assim por diante.

Dado que as instâncias originais de Ropke e Cordeau (2009) assumem frota homogênea e

Tabela 2: Características das instâncias de Ropke e Cordeau (2009).

| Classe | q | w |
|--------|----|-----|
| AA | 15 | 60 |
| BB | 20 | 60 |
| CC | 15 | 120 |
| DD | 20 | 120 |

um único tipo de produto, foi necessário também definir a maneira de geração da frota de veículos heterogêneos e a criação dos múltiplos produtos. Geraram-se duas frotas de veículos com capacidades heterogêneas. A primeira frota, $F1$, é empregada nos testes computacionais onde não se considera a restrição prática de coleta e entrega fracionada. A segunda frota, $F2$, foi concebida e utilizada nos testes computacionais que consideram todas as restrições práticas abordadas neste trabalho (frota heterogênea, múltiplos produtos, coleta e entrega fracionada). A motivação para criação da frota $F2$ é que testes preliminares com o primeiro conjunto de veículos, não apresentaram fracionamento de coleta e entrega para nenhuma das instâncias. Criou-se então, uma frota mais restrita do que a primeira, como descrito a seguir.

A frota $F1$ foi criada gerando-se aleatoriamente o parâmetro q_k entre a maior demanda da classe, definida como $maxD$, e 20% maior do que $maxD$, ou seja, $q_k = [maxD; 1,2.maxD]$. Similarmente, a frota $F2$ foi construída selecionando o parâmetro q_k entre a maior demanda menos 20% e a maior demanda, dessa forma, tem-se $q_k = [0,8.maxD; maxD]$. As capacidades dos veículos para cada frota ($F1$ e $F2$) para as respectivas classes de instâncias podem ser vistas nas Tabelas 3 e 4.

Tabela 3: Frota $F1$ de veículos heterogêneos para as classes de instâncias AA, BB, CC e DD.

| Classes | Capacidades | | | | | | | | |
|---------|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | q_1 | q_2 | q_3 | q_4 | q_5 | q_6 | q_7 | q_8 | q_9 |
| AA/CC | 15 | 18 | 17 | 15 | 15 | 17 | 16 | - | - |
| BB/DD | 23 | 22 | 21 | 20 | 24 | 24 | 20 | 21 | 22 |

O número de produtos criados é três, todos com parâmetro $u_l = 1$. Para cada requisição i de uma instância, a sua respectiva demanda original, $d_i > 0$, é dividida pelo número de produtos, $d_i/|L|$. Caso a divisão seja exata, a quantidade dos produtos passa a ser $d_{il} = d_i/|L|$ para $l = 1, \dots, L$. Se houver resto positivo após a divisão, a demanda de um dos produtos da requisição i é acrescida dessa diferença de forma que $\sum_{l=1}^L d_{il} = d_i$.

Tabela 4: Frota $F2$ de veículos heterogêneos para as classes de instâncias AA, BB, CC e DD.

| Classes | Capacidades | | | | | | | | |
|---------|-------------|-------|-------|-------|-------|-------|-------|-------|-------|
| | q_1 | q_2 | q_3 | q_4 | q_5 | q_6 | q_7 | q_8 | q_9 |
| AA/CC | 15 | 14 | 12 | 14 | 13 | 13 | 12 | - | - |
| BB/DD | 19 | 18 | 18 | 20 | 16 | 17 | 16 | 16 | 19 |

5.2 Calibração dos parâmetros

Todos os parâmetros da ALNS proposta neste trabalho foram calibrados de forma semelhante ao realizado por Ropke e Pisinger (2006). A metodologia usada foi a seguinte: primeiramente todos os parâmetros foram fixados com os valores reportados por Ropke e Pisinger (2006). Em seguida, os parâmetros são calibrados um após o outro, deixando um parâmetro assumir valores pré-estabelecidos, enquanto todos os outros são mantidos fixos. Foram usadas 12 instâncias para a calibração dos parâmetros, sendo três de cada classe (AA, BB, CC e DD), contendo, respectivamente, 30, 40 e 50 requisições. Para cada instância do conjunto usado na calibração, o valor do parâmetro que não está fixado, é escolhido em termos do desvio médio absoluto da melhor solução. Ao final, para o parâmetro que está sendo calibrado, tem-se uma lista de valores que aquele assumiu, e a indicação de quantas vezes cada valor foi escolhido por ter o menor desvio médio em relação à melhor solução encontrada na literatura para cada instância. Dessa forma, é realizada uma média ponderada para escolher o valor do parâmetro em calibração, onde a ponderação é pelo número de vezes que o valor pré-estabelecido foi escolhido como sendo o melhor. O desvio médio absoluto é dado por:

$$\frac{1}{k} \sum_{i=1}^k |v_i - v_{best}|, \quad (5.1)$$

onde k é número de vezes que cada valor pré-estabelecido para um parâmetro em calibração é testado, neste caso $k = 5$, v_i é o valor da função objetivo obtido e v_{best} é o melhor valor reportado na literatura. Os melhores valores para as instâncias usadas na calibração foram retirados do trabalho de Ropke e Cordeau (2009). Esse procedimento é realizado para todo parâmetro. É importante mencionar que os valores reportados pela literatura para as instâncias utilizadas são para o PCEJT clássico, ou seja, com frota homogênea, um único tipo de produto e sem coleta e entrega fracionada. A seguir, apresenta-se na Tabela 5 os parâmetros calibrados e seus respectivos valores, sendo válido relembrá-los. O primeiro parâmetro (w) é usado no cálculo da temperatura inicial do critério de aceite da SA usado na ALNS. Ele determina quanto em porcentagem uma solução qualquer deve ser aceita em relação a uma solução inicial. O segundo parâmetro (c) é o coeficiente de resfriamento da SA. Os parâmetros de número três (σ_1), quatro (σ_2) e cinco (σ_3), são os valores da pontuação atribuída às heurísticas ao final de cada segmento.

O sexto parâmetro (r) é o fator de reação para atualização dos pesos das heurísticas. Já os parâmetros sete (φ), oito (χ) e nove (ψ), são respectivos a heurística de remoção *Shaw Removal* e suas variantes. O décimo parâmetro (δ) é usado para criar aleatoriedade no processo de escolhas de requisições para serem removidas pelas heurísticas de *Shaw Removal* e variantes e pela *Worst Removal*. Por fim, o último parâmetro (η) é usado para cálculo do ruído adicionado na função objetivo das heurísticas de inserção.

Tabela 5: Valores para os parâmetros da ALNS

| Parâmetros | w | c | σ_1 | σ_2 | σ_3 | r | φ | χ | ψ | δ | η |
|-------------------|-----|--------|------------|------------|------------|-----|-----------|--------|--------|----------|--------|
| Valores | 1,3 | 0,9997 | 27 | 30 | 22 | 0,1 | 0,3 | 0,4 | 0,3 | 6 | 0,2 |

5.3 Resultados da ALNS para o PCEJT clássico

Nesta seção, são apresentados os resultados obtidos pela ALNS para o PCEJT clássico e também a comparação destes com aqueles reportados pela literatura. Dessa forma, cada instância foi executada cinco vezes usando a ALNS com um limite de 600 segundos e 25000 iterações. Os valores para as soluções obtidas pela ALNS proposta neste trabalho foram comparados com aqueles obtidos pela implementação da ALNS proposta por Ropke e Cordeau (2009) e pela formulação apresentada por Furtado et al. (2017). Foi necessário utilizar dois trabalhos de comparação, pois o primeiro apresenta resultados para instâncias contendo de 30 a 75 requisições. As instâncias contendo de cinco até 25 requisições foram introduzidas em trabalho um posterior, por Furtado et al. (2017), como já mencionado na seção anterior. Os resultados da meta-heurística para o PCEJT clássico podem ser observados pela Tabela 6. A primeira coluna apresenta as instâncias utilizadas, e a segunda coluna apresenta a melhor solução encontrada pela ALNS proposta neste trabalho. A terceira coluna apresenta o tempo para a melhor solução encontrada da ALNS. A melhor solução reportada pela literatura bem como o melhor tempo para estas soluções é apresentada na quarta e quinta coluna, respectivamente. Por fim, na sexta coluna apresenta-se o *Gap* entre as soluções encontradas pela ALNS e a melhor solução da literatura. O referido *Gap* é calculado pela expressão (5.2), onde $ALNS_{sol}$ é a melhor solução encontrada pela ALNS e Lit_{sol} é a melhor solução reportada na literatura.

$$Gap = \left\{ \frac{ALNS_{sol} - Lit_{sol}}{Lit_{sol}} \right\} .100. \quad (5.2)$$

Tabela 6: Comparação entre resultados da meta-heurística e literatura.

| Instâncias | Melhor Sol. ALNS | Melhor T. ALNS | Melhor Sol. Lit. | Melhor T. Lit. | Gap Sol. Lit. |
|------------|------------------|----------------|------------------|----------------|---------------|
| AA5 | 10184,80 | 0,00 | 10184,80 | 0,40 | 0,000% |
| AA10 | 10383,60 | 0,00 | 10383,60 | 1,42 | 0,000% |
| AA15 | 20542,40 | 9,00 | 20542,40 | 8,18 | 0,000% |
| AA20 | 20759,30 | 30,00 | 20759,40 | 41,00 | 0,000% |
| AA25 | 21055,30 | 9,00 | 21055,30 | 2,00 | 0,000% |
| AA30 | 31119,10 | 61,00 | 31119,10 | 75,00 | 0,000% |
| AA35 | 31299,70 | 156,00 | 31299,80 | 93,00 | 0,000% |
| AA40 | 31518,10 | 309,00 | 31515,90 | 62,00 | 0,007% |
| AA45 | 41649,20 | 322,00 | 31759,80 | 124,00 | 31,138% |
| AA50 | 41775,70 | 523,00 | 41775,00 | 159,00 | 0,002% |
| AA55 | 41925,10 | 548,00 | 41907,80 | 185,00 | 0,041% |
| AA60 | 42194,40 | 553,00 | 42140,70 | 118,00 | 0,127% |
| AA65 | 42305,80 | 581,00 | 42252,70 | 135,00 | 0,126% |
| AA70 | 52519,90 | 596,00 | 42455,00 | 187,00 | 23,707% |
| AA75 | 52653,30 | 481,00 | 52472,70 | 308,00 | 0,344% |
| BB5 | 10339,70 | 0,00 | 10339,70 | 0,12 | 0,000% |
| BB10 | 20512,40 | 0,00 | 20512,50 | 0,36 | 0,000% |
| BB15 | 20667,50 | 0,00 | 20667,60 | 0,45 | 0,000% |
| BB20 | 20791,30 | 1,00 | 20791,30 | 0,64 | 0,000% |
| BB25 | 20987,90 | 0,00 | 20988,00 | 2,33 | 0,000% |
| BB30 | 31086,30 | 67,00 | 31086,30 | 76,00 | 0,000% |
| BB35 | 31281,70 | 89,00 | 31281,20 | 92,00 | 0,002% |
| BB40 | 31493,30 | 176,00 | 31493,40 | 63,00 | 0,000% |
| BB45 | 41571,40 | 187,00 | 41551,10 | 127,00 | 0,049% |
| BB50 | 41704,70 | 413,00 | 41703,40 | 160,00 | 0,003% |
| BB55 | 41885,70 | 313,00 | 41885,70 | 96,00 | 0,000% |
| BB60 | 62430,60 | 586,00 | 62420,10 | 178,00 | 0,017% |
| BB65 | 62688,20 | 587,00 | 62639,10 | 202,00 | 0,078% |
| BB70 | 73045,90 | 597,00 | 62952,30 | 236,00 | 16,034% |
| BB75 | 73175,50 | 441,00 | 63130,40 | 256,00 | 15,912% |
| CC5 | 10212,50 | 0,00 | 10212,50 | 0,17 | 0,000% |
| CC10 | 20385,50 | 0,00 | 20385,50 | 0,45 | 0,000% |
| CC15 | 20571,10 | 2,00 | 20571,20 | 2,77 | 0,000% |
| CC20 | 20764,40 | 3,00 | 20764,40 | 25,90 | 0,000% |
| CC25 | 20944,30 | 17,00 | 20944,40 | 136,60 | 0,000% |
| CC30 | 31087,70 | 57,00 | 31087,70 | 76,00 | 0,000% |
| CC35 | 31230,60 | 139,00 | 31230,60 | 97,00 | 0,000% |
| CC40 | 31361,00 | 201,00 | 31358,50 | 132,00 | 0,008% |
| CC45 | 31509,10 | 373,00 | 31509,10 | 82,00 | 0,000% |
| CC50 | 41694,50 | 398,00 | 41689,00 | 168,00 | 0,013% |
| CC55 | 41843,60 | 372,00 | 41836,30 | 196,00 | 0,017% |

continua

Tabela 6 – Resultados para a meta-heurística - continuação

| Instâncias | Melhor Sol. ALNS | Melhor T. ALNS | Melhor Sol. Lit. | Melhor T. Lit. | Gap Sol. Lit. |
|-------------------|-------------------------|-----------------------|-------------------------|-----------------------|----------------------|
| CC60 | 42035,20 | 493,00 | 42015,50 | 127,00 | 0,047% |
| CC65 | 52188,50 | 594,00 | 42172,10 | 145,00 | 23,751% |
| CC70 | 52351,30 | 573,00 | 52201,90 | 288,00 | 0,286% |
| CC75 | 52583,70 | 190,00 | 52375,60 | 325,00 | 0,397% |
| DD5 | 10324,80 | 0,00 | 10324,80 | 0,12 | 0,000% |
| DD10 | 20567,80 | 0,00 | 20567,80 | 0,91 | 0,000% |
| DD15 | 20673,30 | 1,00 | 20673,40 | 18,70 | 0,000% |
| DD20 | 20776,20 | 30,00 | 20776,20 | 47,80 | 0,000% |
| DD25 | 21004,90 | 46,00 | 21004,90 | 6158,20 | 0,000% |
| DD30 | 21133,30 | 43,00 | 21133,30 | 49,00 | 0,000% |
| DD35 | 31228,90 | 145,00 | 31224,00 | 99,00 | 0,016% |
| DD40 | 31352,10 | 225,00 | 31352,20 | 136,00 | 0,000% |
| DD45 | 31495,40 | 323,00 | 31483,90 | 132,00 | 0,037% |
| DD50 | 31613,30 | 478,00 | 31600,90 | 105,00 | 0,039% |
| DD55 | 31789,80 | 546,00 | 31743,30 | 124,00 | 0,146% |
| DD60 | 41893,50 | 568,00 | 41869,40 | 274,00 | 0,058% |
| DD65 | 42208,80 | 544,00 | 42125,70 | 209,00 | 0,197% |
| DD70 | 42351,90 | 579,00 | 42220,30 | 175,00 | 0,312% |
| DD75 | 52538,30 | 593,00 | 42396,80 | 201,00 | 23,920% |

Analisando a Tabela 6 é possível identificar que a meta-heurística proposta por essa pesquisa encontrou 30 soluções, de um total de 60 instâncias, cujo os valores da função objetivo são iguais aos apresentados pelos trabalhos de Furtado et al. (2017) e Ropke e Cordeau (2009). Das 30 instâncias remanescentes, a ALNS encontrou 14 soluções em que o valor da função objetivo tem um *Gap* igual ou maior que 0,1%. Sendo que destas 14, somente para seis instâncias foram obtidas soluções em que o *Gap* de suas funções objetivos foram maiores que 1% do que os valores reportados pelo trabalho de Ropke e Cordeau (2009). É importante mencionar que para estas, a ALNS proposta nesta pesquisa encontrou soluções com um veículo a mais do que os resultados apresentados na literatura de referência. *Gaps* inferiores 0,1% foram obtidos para 16 instâncias comparados com os resultados de Ropke e Cordeau (2009). As Figuras 4, 5, 6 e 7 ilustram o comportamento dos resultados dos testes conduzidos.

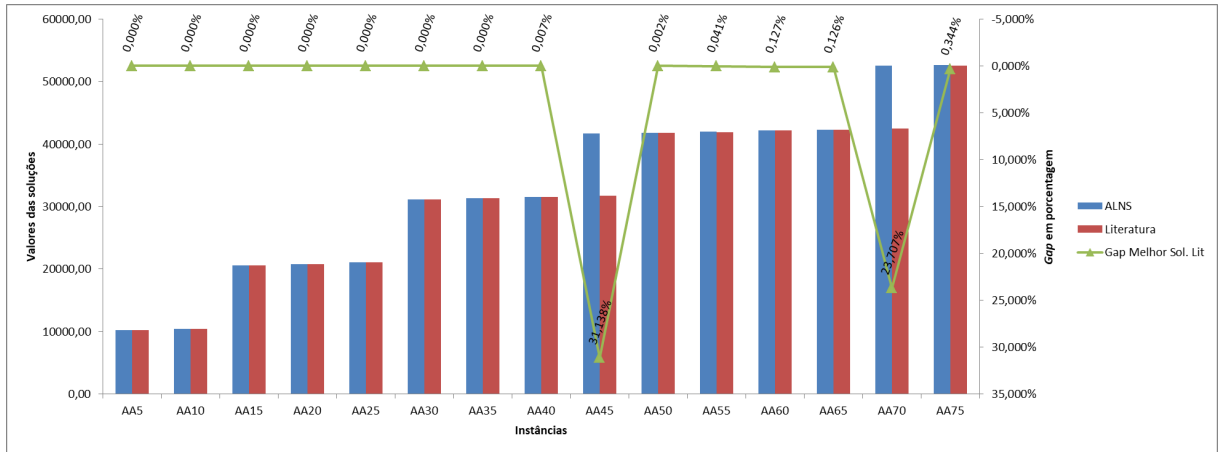


Figura 4: Comparação entre soluções ALNS e Literatura para a classe AA.

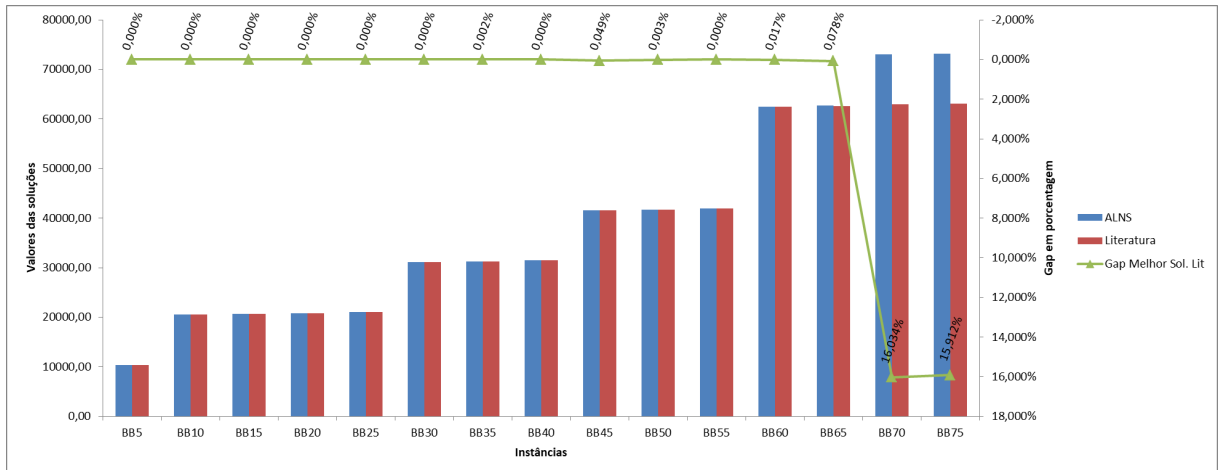


Figura 5: Comparação entre soluções ALNS e Literatura para a classe BB.

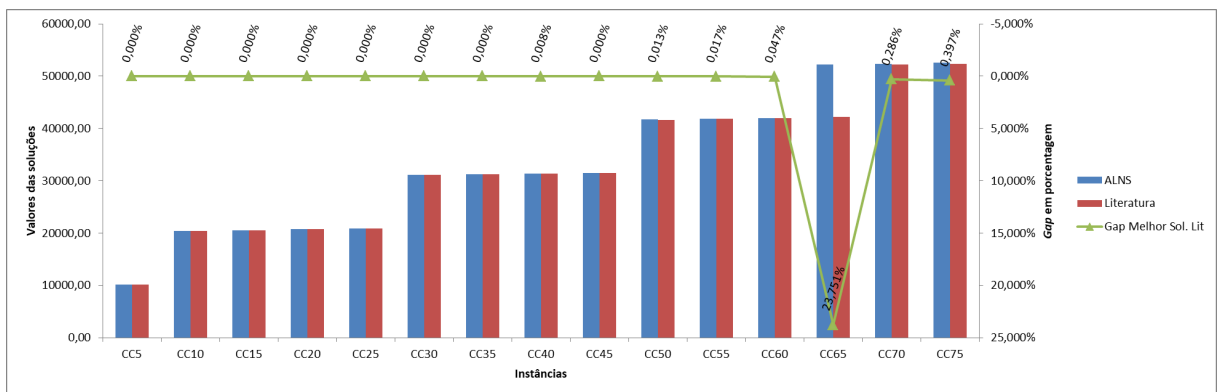


Figura 6: Comparação entre soluções ALNS e Literatura para a classe CC.

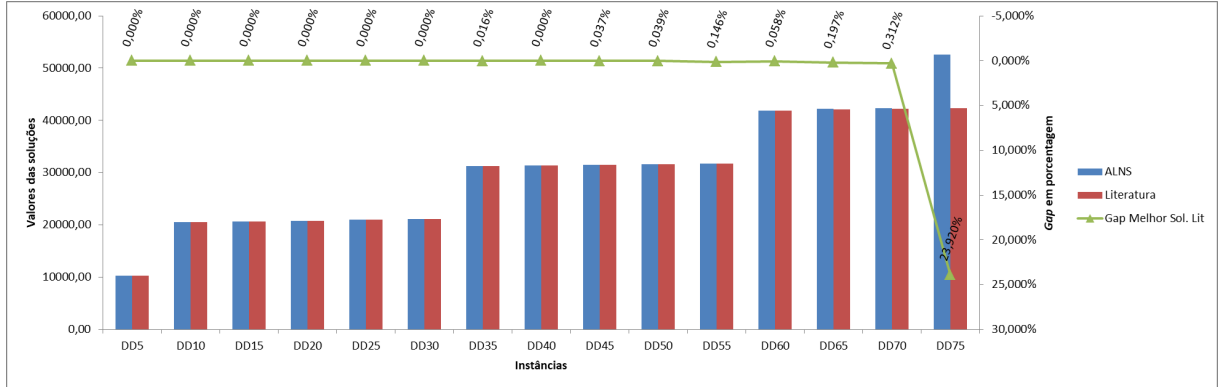


Figura 7: Comparação entre soluções ALNS e Literatura para a classe DD.

Avaliando-se a implementação da ALNS proposta, pode-se dizer que a mesma teve desempenho aceitável para as instâncias utilizadas para o PCEJT clássico. Embora seja notório que a mesma pode ser melhorada tanto no que tange a valores das soluções quanto a tempos gastos para obtê-las, a implementação apresentada neste trabalho é passível de ser adaptada para os casos particulares estudados como para aplicações reais.

5.4 Resultados para o PCEJT-1

Nesta seção, são apresentados os resultados para o PCEJT-1, tanto para os modelos de três- e dois-índices como para a ALNS. Os resultados dos modelos de três- e dois-índices para as instâncias usadas são apresentados na Tabela 7 considerando-se um limite de tempo de 3600 segundos. A primeira coluna apresenta o nome da instância e posteriormente, para cada modelo, as colunas representam respectivamente o valor da função objetivo (UB), o *Gap* correspondente e o tempo computacional em segundos (Tempo). Reportam-se apenas os resultados que tiveram o *Gap* disponível pelo CPLEX, para pelo menos um dos modelos. O *Gap* (relativo) é calculado pelo CPLEX usando o valor da função objetivo $f(x)$ e de um limite inferior lb para a solução do problema, conforme definido por (5.3). Para todas as outras instâncias não foram encontradas soluções dentro do limite de tempo estabelecido. Caso os tempos das soluções obtidas tenha atingido o máximo permitido, reporta-se apenas como TL.

$$Gap = \frac{f(x) - lb}{lb}. \quad (5.3)$$

Tabela 7: Resultados para os modelos de três- e dois-índices.

| Instâncias | Modelo três-índices | | | Modelo dois-índices | | |
|------------|---------------------|--------|-------|---------------------|--------|-----------|
| | UB | Gap | Tempo | UB | Gap | Tempo (s) |
| AA5 | 10184,80 | 0,00% | 0,17 | 10184,80 | 0,00% | 0,08 |
| AA10 | 10383,60 | 0,00% | 2,70 | 10383,60 | 0,00% | 0,41 |
| AA15 | 20542,40 | 0,00% | TL | 20542,40 | 0,00% | 11,72 |
| AA20 | 20762,90 | 1,00% | TL | 20757,90 | *0,00% | TL |
| AA25 | - | - | - | 30976,20 | 33,00% | TL |
| BB5 | 10339,70 | 0,00% | 0,13 | 10339,70 | 0,00% | 0,08 |
| BB10 | 20512,50 | 0,00% | 25,47 | 20512,50 | 0,00% | 0,92 |
| BB15 | 20667,60 | 0,00% | TL | 20667,60 | 0,00% | 15,86 |
| BB20 | - | - | - | 20786,00 | 0,00% | 106,52 |
| BB25 | - | - | - | 20946,10 | 1,00% | TL |
| CC5 | 10212,50 | 0,00% | 5,61 | 10212,50 | 0,00% | 0,22 |
| CC10 | 10394,30 | 1,00% | TL | 10394,30 | 0,00% | 6,49 |
| CC15 | - | - | - | 20557,80 | 49,00% | TL |
| CC20 | - | - | - | 20748,90 | 49,00% | TL |
| CC25 | - | - | - | - | - | - |
| DD5 | 10324,80 | 0,00% | 0,28 | 10324,80 | 0,00% | 0,09 |
| DD10 | 20567,80 | 49,00% | TL | 20567,80 | 0,00% | 41,49 |
| DD15 | - | - | - | 20616,40 | 49,00% | TL |
| DD20 | - | - | - | 20679,20 | 49,00% | TL |
| DD25 | - | - | - | - | - | - |

*esta solução não é ótima de acordo com a precisão do CPLEX.

É possível observar que o modelo de dois-índices é superior ao de três-índices, encontrando soluções para mais instâncias, para todas as classes de problemas. Esses resultados corroboram com os reportados por Furtado et al. (2017) sobre a eficiência obtida com o uso desta formulação frente a outras da literatura para o PCEJT. Observa-se também, que as maiores instâncias com resultados reportados para ambos os modelos de três- e dois-índices são instâncias com 20 e 25 requisições, respectivamente. Dessa forma, instâncias contendo 30 requisições ou mais não tiveram qualquer solução reportada pelos modelos dentro de 3600 segundos.

Para a meta-heurística, os resultados podem ser observados na Tabela 8. Cada instância foi executada cinco vezes usando a ALNS com um limite de tempo de 600 segundos e 25000 iterações. A primeira coluna identifica as instâncias, a segunda apresenta a média das soluções calculada a partir das soluções obtidas das cinco execuções. A terceira coluna apresenta o coeficiente de variação como medida de dispersão das soluções obtidas, o qual é calculado dividindo-se o desvio padrão pela média destas soluções. O coeficiente de variação demonstra a estabilidade da ALNS ao encontrar as soluções. A quarta coluna apresenta a média dos tempos de execução da meta-heurística. A melhor solução e o respectivo tempo transcorrido até obtê-la pela ALNS

são vistos nas colunas cinco e seis.

Tabela 8: Resultados para a meta-heurística.

| Instâncias | Média S. | Coeficiente V. | Média T. | Melhor S. | Tempo Melhor S. |
|-------------------|-----------------|-----------------------|-----------------|------------------|------------------------|
| AA5 | 10184,80 | 0,00 | 0,00 | 10184,80 | 0,00 |
| AA10 | 10383,60 | 0,00 | 0,00 | 10383,60 | 0,00 |
| AA15 | 20542,40 | 0,00 | 1,80 | 20542,40 | 0,00 |
| AA20 | 20757,90 | 0,00 | 17,20 | 20757,90 | 0,00 |
| AA25 | 21041,08 | *0,00 | 4,20 | 21041,00 | 1,00 |
| AA30 | 31117,20 | *0,00 | 126,00 | 31116,00 | 136,00 |
| AA35 | 31279,10 | *0,00 | 286,20 | 31277,10 | 182,00 |
| AA40 | 31448,90 | *0,00 | 272,60 | 31448,70 | 195,00 |
| AA45 | 31673,46 | *0,00 | 404,20 | 31666,60 | 445,00 |
| AA50 | 41770,50 | *0,00 | 475,80 | 41733,80 | 487,00 |
| AA55 | 41939,18 | *0,00 | 540,20 | 41930,90 | 436,00 |
| AA60 | 42133,80 | *0,00 | 410,60 | 42108,40 | 599,00 |
| AA65 | 42268,96 | *0,00 | 312,60 | 42238,60 | 449,00 |
| AA70 | 52491,64 | *0,00 | 405,00 | 52448,50 | 150,00 |
| AA75 | 52625,50 | *0,00 | 341,20 | 52575,10 | 495,00 |
| BB5 | 10339,70 | 0,00 | 0,00 | 10339,70 | 0,00 |
| BB10 | 20512,50 | 0,00 | 0,00 | 20512,50 | 0,00 |
| BB15 | 20667,60 | 0,00 | 0,20 | 20667,60 | 0,00 |
| BB20 | 20786,00 | 0,00 | 1,80 | 20786,00 | 0,00 |
| BB25 | 20947,40 | *0,00 | 6,40 | 20946,10 | 0,00 |
| BB30 | 31038,14 | *0,00 | 127,20 | 31036,70 | 127,00 |
| BB35 | 31228,68 | *0,00 | 137,00 | 31225,00 | 174,00 |
| BB40 | 31449,06 | *0,00 | 177,00 | 31447,30 | 46,00 |
| BB45 | 41535,02 | *0,00 | 447,40 | 41522,40 | 450,00 |
| BB50 | 41681,20 | *0,00 | 511,20 | 41652,60 | 513,00 |
| BB55 | 41866,54 | *0,00 | 462,00 | 41853,70 | 555,00 |
| BB60 | 62361,98 | *0,00 | 579,40 | 62344,20 | 583,00 |
| BB65 | 62673,42 | *0,00 | 498,60 | 62635,20 | 596,00 |
| BB70 | 62987,24 | *0,00 | 356,40 | 62920,80 | 544,00 |
| BB75 | 71104,30 | 0,06 | 280,00 | 63031,10 | 175,00 |
| CC5 | 10212,50 | *0,00 | 0,00 | 10212,50 | 0,00 |
| CC10 | 10394,30 | 0,00 | 0,00 | 10394,30 | 0,00 |
| CC15 | 20554,90 | 0,00 | 0,80 | 20554,90 | 0,00 |
| CC20 | 20721,40 | 0,00 | 20,40 | 20721,40 | 2,00 |
| CC25 | 20908,80 | 0,00 | 64,80 | 20908,80 | 38,00 |
| CC30 | 31036,02 | *0,00 | 94,00 | 31010,20 | 93,00 |
| CC35 | 31168,04 | *0,00 | 270,20 | 31105,00 | 576,00 |
| CC40 | 31283,42 | *0,00 | 294,80 | 31276,40 | 335,00 |
| CC45 | 31430,64 | *0,00 | 351,00 | 31419,70 | 423,00 |
| CC50 | 41671,38 | *0,00 | 492,80 | 41628,80 | 505,00 |

continua

Tabela 8 – Resultados para a meta-heurística - continuação

| Instâncias | Média S. | Coefficiente V. | Média T. | Melhor S. | Tempo Melhor S. |
|-------------------|-----------------|------------------------|-----------------|------------------|------------------------|
| CC55 | 41771,42 | *0,00 | 551,00 | 41745,00 | 523,00 |
| CC60 | 42012,26 | *0,00 | 423,80 | 41997,80 | 438,00 |
| CC65 | 52196,38 | *0,00 | 452,60 | 52147,60 | 562,00 |
| CC70 | 52382,60 | *0,00 | 433,40 | 52353,60 | 534,00 |
| CC75 | 52564,38 | *0,00 | 364,00 | 52519,80 | 369,00 |
| DD5 | 10324,80 | 0,00 | 0,00 | 10324,80 | 0,00 |
| DD10 | 20567,80 | 0,00 | 0,00 | 20567,80 | 0,00 |
| DD15 | 20616,40 | 0,00 | 0,00 | 20616,40 | 0,00 |
| DD20 | 20679,20 | 0,00 | 15,80 | 20679,20 | 11,00 |
| DD25 | 20902,84 | *0,00 | 84,00 | 20902,30 | 66,00 |
| DD30 | 21016,68 | *0,00 | 113,60 | 21015,10 | 65,00 |
| DD35 | 31095,08 | *0,00 | 209,40 | 31083,50 | 201,00 |
| DD40 | 31209,54 | *0,00 | 260,60 | 31202,20 | 280,00 |
| DD45 | 31326,74 | *0,00 | 413,20 | 31310,90 | 442,00 |
| DD50 | 31444,38 | *0,00 | 508,80 | 31430,20 | 536,00 |
| DD55 | 31620,22 | *0,00 | 484,20 | 31581,60 | 515,00 |
| DD60 | 41794,32 | *0,00 | 549,00 | 41762,30 | 513,00 |
| DD65 | 42145,56 | *0,00 | 488,60 | 42129,80 | 442,00 |
| DD70 | 42249,50 | *0,00 | 330,20 | 42228,20 | 446,00 |
| DD75 | 50446,74 | 0,09 | 485,60 | 42462,70 | 478,00 |

*coeficiente de variação não nulo e menor que 0,01.

Pode-se observar pela Tabela 8, que a ALNS obteve soluções factíveis dentro do tempo estipulado de 600 segundos para todas as instâncias. É possível notar que o coeficiente de variação para a média das soluções é muito próximo ou até mesmo zero para a maioria das instâncias, com exceção das instâncias BB75 e DD75 que obtiveram coeficiente de variação 0,06 e 0,09 respectivamente. Esse fato decorre da composição das soluções usadas para o cálculo da média observada na segunda coluna. Nessa composição, encontra-se uma solução com menor número de veículos encontrado pela meta-heurística. Os tempos médios e o melhor tempo para a ALNS encontrar uma solução apresentam uma tendência de aumento, conforme as instâncias ficam maiores em termos do número de requisições. No entanto, evidencia-se que para as instâncias AA20, AA25, AA70, BB40 e CC20 a melhor solução foi encontrada em um tempo inferior à metade do tempo médio levado pela meta-heurística nas outras execuções. Uma hipótese é que esse fato decorra de uma melhor solução inicial fornecida pela heurística construtiva. Outra hipótese seria uma maior participação de determinada heurística de remoção em relação as outras.

Para a meta-heurística, as melhores soluções, o *Gap* da média para a melhor solução e a média das execuções para as classes de instâncias AA, BB, CC e DD são representados pelas Figuras 8, 9, 10 e 11 respectivamente. O *Gap* é aqui calculado semelhantemente à equação (5.2),

no entanto, $ALNS_{sol}$ refere-se à média das soluções e Lit_{sol} é pertinente à melhor solução obtida dentre as cinco execuções. Por fim, ao analisarmos as figuras é possível observar a discrepância entre a média das soluções obtidas e a melhor solução encontrada pela ALNS, para as instâncias BB75 e DD75, enquanto que para todas as outras, os *Gaps* foram menores que 1%. Outro aspecto importante é percebido ao contrapor-se os *Gaps* aos valores dos coeficientes de variação. Ao fazê-lo pode-se inferir que existe certa estabilidade no processo de busca da meta-heurística proposta e que o mesmo não é totalmente aleatório.

Uma comparação entre a ALNS e os modelos de dois- e três-índices é demonstrada pela Figura 12 considerando apenas instâncias que tiveram alguma solução reportada por ao menos um dos modelos. A escala do eixo dos tempos teve o máximo valor estabelecido em 30 segundos. Essa definição é devido à dificuldade de representação no gráfico dos valores referentes aos testes dos modelos, que para determinadas instâncias, tiveram tempos computacionais relativamente altos. Somente os rótulos dos tempos computacionais dos modelos foram representados.

Observando os resultados obtidos pelos modelos e pela meta-heurística proposta, aqueles têm desempenho melhor para instâncias pequenas de até 15 requisições obtendo soluções ótimas. Já a ALNS consegue obter bons resultados para todas as instâncias dentro de tempos computacionais aceitáveis. Por exemplo, para a instância AA25, o modelo de três-índices não achou solução dentro do tempo estipulado, enquanto o modelo de dois-índices conseguiu encontrar uma solução factível com um *Gap* de aproximadamente 30% contendo três veículos, sendo que a ALNS conseguiu obter uma solução utilizando dois veículos em um tempo desprezível.

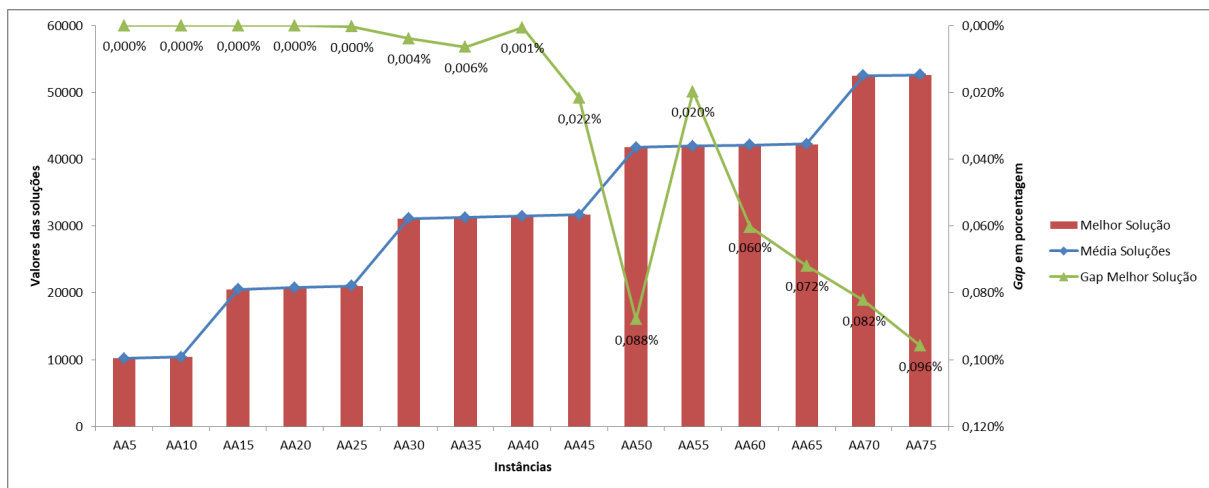


Figura 8: Soluções para a classe AA.

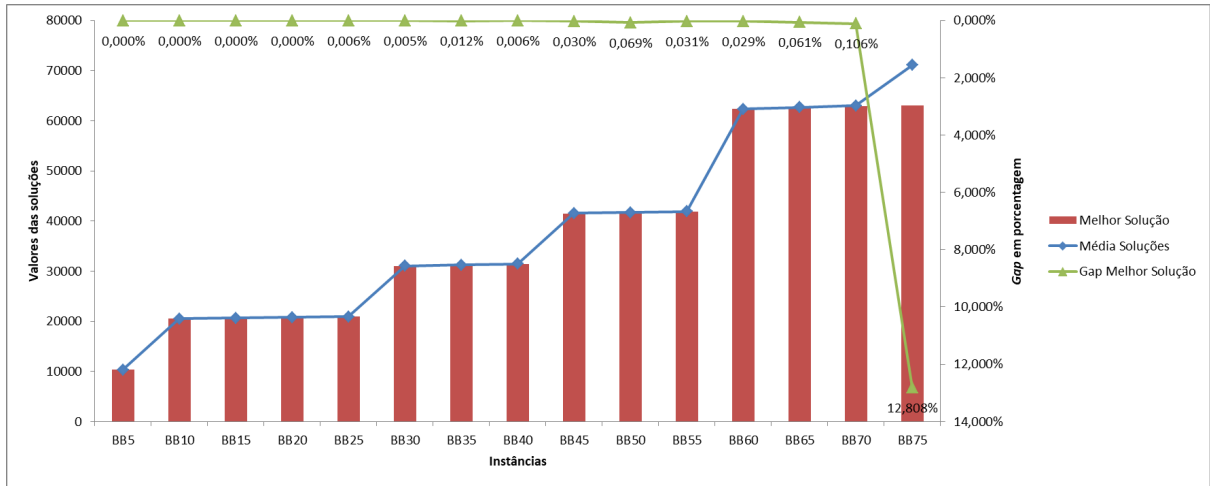


Figura 9: Soluções para a classe BB.

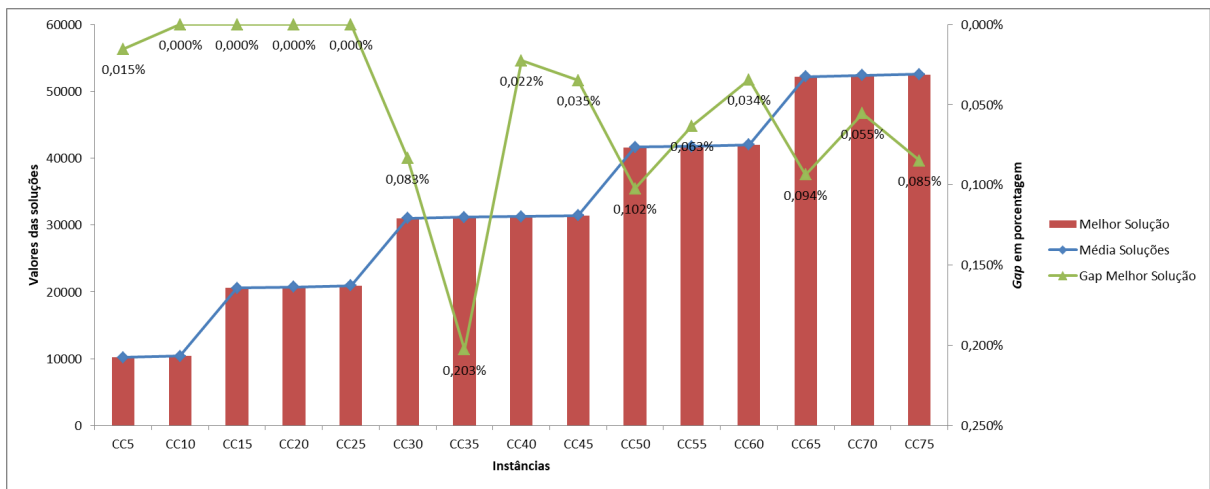


Figura 10: Soluções para a classe CC.

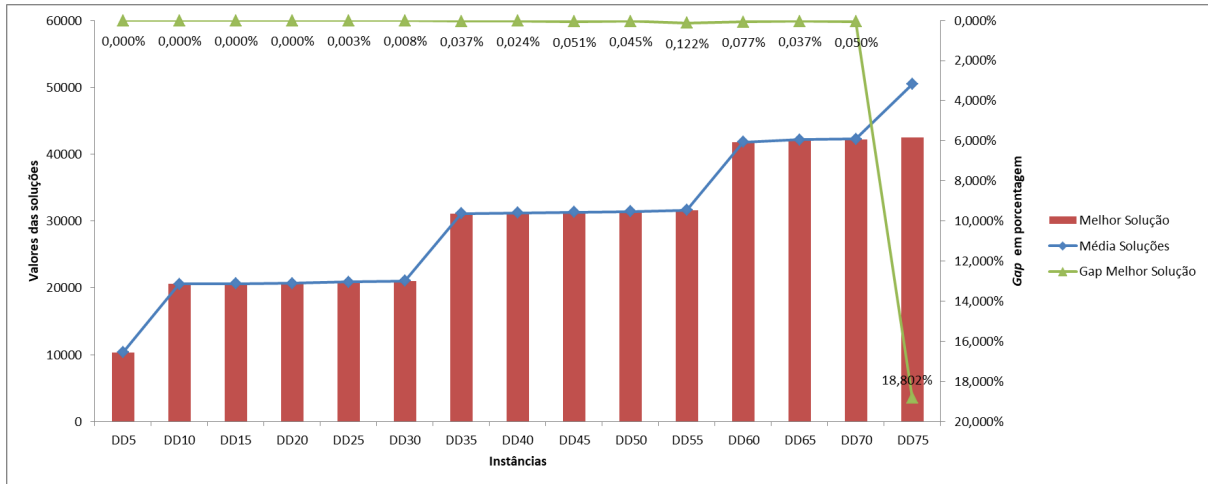


Figura 11: Soluções para a classe DD.

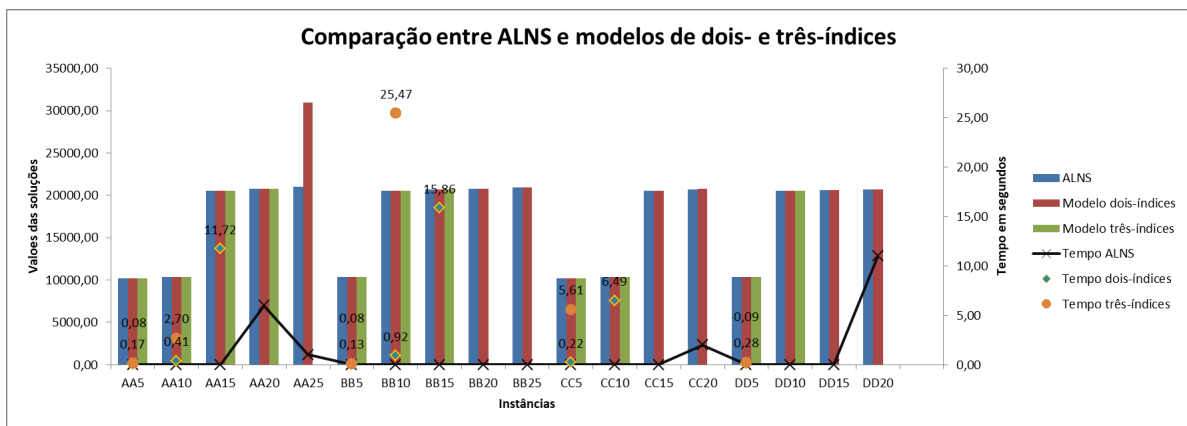


Figura 12: Comparação entre ALNS e modelos de dois- e três-índices.

5.5 Resultados para o PCEJT-2

Nesta seção, apresentam-se os resultados para o modelo de três-índices e para a meta-heurística, contendo restrições de frota heterogênea, múltiplos produtos e coleta e entrega fracionada. A interpretação das Tabelas 9 e 10 é a mesma da seção anterior. A seguir, apresentam-se os resultados para o modelo de três-índices, considerando apenas instâncias que tiveram uma solução reportada pelo CPLEX dentro do tempo limite estabelecido.

Pela análise da Tabela 9, identifica-se que o modelo de três-índices encontrou soluções para 12 instâncias da literatura de um total de 60. Observa-se que sete destes resultados são ótimos. É possível observar que o modelo obteve melhor desempenho para a classe de instâncias AA, pois encontrou três soluções ótimas de um total de quatro. O modelo obteve desempenho equiparável

Tabela 9: Resultados para o modelo de três-índices com coleta e entrega fracionada.

| Instâncias | UB | Gap | Tempo(s) |
|-------------------|-----------|------------|-----------------|
| AA5 | 10184,80 | 0,00% | 0,30 |
| AA10 | 10383,60 | 0,00% | 13,75 |
| AA15 | 20544,70 | 0,00% | 2028,91 |
| AA20 | 30778,80 | 64,00% | TL |
| BB5 | 10339,70 | 0,00% | 0,14 |
| BB10 | 20525,40 | 0,00% | 8,16 |
| BB15 | 30721,60 | 33,00% | TL |
| BB20 | 40957,40 | 50,00% | TL |
| CC5 | 10212,50 | 0,00% | 27,63 |
| CC10 | 20393,30 | 49,00% | TL |
| DD5 | 10324,80 | 0,00% | 1,33 |
| DD10 | 20588,70 | 48,00% | TL |

em relação ao número de instâncias solucionadas para a classe BB, embora o número de soluções ótimas obtidas seja a metade daquele observado ao resolver-se a classe AA. Para ambas as classes AA e BB o modelo achou soluções para instâncias com até 20 requisições. Outro ponto a ressaltar para as instâncias das classes AA e BB que tiveram soluções reportadas, é que o modelo esgotou o tempo computacional sem encontrar uma solução ótima para somente uma instância da classe AA e para duas da classe BB. Para as classes CC e DD o modelo obteve soluções para instâncias com até 10 requisições. No entanto, para ambas, o *Gap* obtido para instâncias com 10 requisições foi próximo de 50%, isso após esgotado o tempo computacional de 3600 segundos.

Os testes computacionais da meta-heurística para as instâncias seguem os mesmos procedimentos descritos na Seção 5.4. A Tabela 10 apresenta os resultados para a ALNS.

Tabela 10: Resultados para a meta-heurística com coleta e entrega fracionada.

| Instâncias | Média S. | Coefficiente V. | Média T. | Melhor S. | Tempo Melhor S. |
|-------------------|-----------------|------------------------|-----------------|------------------|------------------------|
| AA5 | 10184,84 | 0,00 | 0,00 | 10184,84 | 0,00 |
| AA10 | 10383,62 | 0,00 | 0,00 | 10383,62 | 0,00 |
| AA15 | 20544,67 | 0,00 | 2,00 | 20544,67 | 4,00 |
| AA20 | 20764,26 | 0,00 | 8,40 | 20764,26 | 5,00 |
| AA25 | 30987,84 | *0,00 | 85,00 | 30986,92 | 57,00 |
| AA30 | 31121,34 | *0,00 | 125,80 | 31121,16 | 138,00 |
| AA35 | 31308,36 | *0,00 | 175,20 | 31303,82 | 96,00 |
| AA40 | 31562,34 | *0,00 | 280,60 | 31551,35 | 327,00 |
| AA45 | 41715,71 | *0,00 | 397,60 | 41701,65 | 365,00 |
| AA50 | 41876,04 | *0,00 | 542,80 | 41850,15 | 596,00 |
| AA55 | 42098,23 | *0,00 | 427,20 | 42087,35 | 586,00 |

continua

Tabela 10 – Resultados para a meta-heurística com coleta e entrega fracionada - continuação

| Instâncias | Média S. | Coefficiente V. | Média T. | Melhor S. | Tempo Melhor S. |
|------------|----------|-----------------|----------|-----------|-----------------|
| AA60 | 52317,29 | *0,00 | 463,40 | 52290,98 | 472,00 |
| AA65 | 52477,20 | *0,00 | 492,40 | 52446,23 | 566,00 |
| AA70 | 52688,43 | *0,00 | 253,00 | 52667,81 | 261,00 |
| AA75 | 52823,34 | *0,00 | 277,00 | 52808,04 | 482,00 |
| BB5 | 10339,67 | 0,00 | 0,00 | 10339,67 | 0,00 |
| BB10 | 20525,43 | 0,00 | 0,00 | 20525,43 | 0,00 |
| BB15 | 20737,32 | 0,00 | 0,60 | 20737,32 | 1,00 |
| BB20 | 30870,56 | *0,00 | 4,60 | 30870,56 | 1,00 |
| BB25 | 31031,07 | 0,00 | 16,20 | 31031,07 | 5,00 |
| BB30 | 31164,12 | *0,00 | 60,80 | 31162,69 | 63,00 |
| BB35 | 39399,61 | 0,11 | 155,20 | 31455,34 | 14,00 |
| BB40 | 41550,23 | *0,00 | 321,40 | 41512,71 | 354,00 |
| BB45 | 45655,58 | 0,12 | 433,20 | 41607,57 | 532,00 |
| BB50 | 49890,57 | 0,17 | 422,80 | 41893,83 | 250,00 |
| BB55 | 64116,78 | 0,07 | 364,00 | 62092,37 | 350,00 |
| BB60 | 80759,21 | 0,06 | 391,20 | 72753,79 | 87,00 |
| BB65 | 81057,89 | 0,06 | 320,40 | 72973,82 | 48,00 |
| BB70 | 83353,63 | 0,09 | 450,80 | 73294,68 | 578,00 |
| BB75 | 83487,35 | 0,08 | 401,80 | 73448,66 | 508,00 |
| CC5 | 10212,53 | 0,00 | 0,00 | 10212,53 | 0,00 |
| CC10 | 20385,50 | 0,00 | 0,00 | 20385,50 | 0,00 |
| CC15 | 20571,18 | 0,00 | 2,60 | 20571,18 | 1,00 |
| CC20 | 20764,42 | 0,00 | 3,80 | 20764,42 | 7,00 |
| CC25 | 20965,74 | *0,00 | 40,00 | 20961,74 | 81,00 |
| CC30 | 31149,34 | *0,00 | 87,60 | 31128,32 | 99,00 |
| CC35 | 31285,71 | *0,00 | 237,40 | 31272,88 | 365,00 |
| CC40 | 31417,45 | *0,00 | 310,40 | 31415,43 | 317,00 |
| CC45 | 33569,02 | 0,13 | 303,20 | 31561,76 | 342,00 |
| CC50 | 41775,38 | *0,00 | 486,40 | 41765,28 | 594,00 |
| CC55 | 41991,89 | *0,00 | 467,00 | 41976,97 | 470,00 |
| CC60 | 52217,06 | *0,00 | 533,80 | 52199,99 | 535,00 |
| CC65 | 52400,97 | *0,00 | 369,00 | 52365,03 | 538,00 |
| CC70 | 52599,09 | *0,00 | 462,40 | 52567,34 | 578,00 |
| CC75 | 62847,57 | *0,00 | 289,20 | 62817,20 | 525,00 |
| DD5 | 10324,77 | 0,00 | 0,00 | 10324,77 | 0,00 |
| DD10 | 20572,97 | 0,00 | 0,20 | 20572,97 | 0,00 |
| DD15 | 20680,33 | 0,00 | 1,80 | 20680,33 | 0,00 |
| DD20 | 20777,65 | *0,00 | 26,80 | 20776,23 | 71,00 |
| DD25 | 21009,28 | 0,00 | 122,00 | 21009,28 | 119,00 |
| DD30 | 21150,37 | *0,00 | 104,20 | 21137,71 | 138,00 |
| DD35 | 31254,80 | *0,00 | 224,40 | 31250,76 | 259,00 |

continua

Tabela 10 – Resultados para a meta-heurística com coleta e entrega fracionada - continuação

| Instâncias | Média S. | Coefficiente V. | Média T. | Melhor S. | Tempo Melhor S. |
|------------|----------|-----------------|----------|-----------|-----------------|
| DD40 | 31387,26 | *0,00 | 372,60 | 31382,67 | 516,00 |
| DD45 | 31512,85 | *0,00 | 449,20 | 31510,66 | 293,00 |
| DD50 | 31643,95 | *0,00 | 534,00 | 31628,68 | 543,00 |
| DD55 | 39829,68 | 0,11 | 486,60 | 31905,41 | 311,00 |
| DD60 | 42053,71 | *0,00 | 491,40 | 42022,93 | 506,00 |
| DD65 | 46351,74 | 0,12 | 531,40 | 42341,40 | 438,00 |
| DD70 | 52574,08 | *0,00 | 284,80 | 52560,46 | 347,00 |
| DD75 | 52764,83 | *0,00 | 460,80 | 52743,20 | 544,00 |

*coeficiente de variação não nulo e menor que 0,01.

Analisando a Tabela 10, é possível perceber que a meta-heurística encontrou soluções factíveis para todas as classes de instâncias. Constata-se para as instâncias BB35, BB45 a BB75, CC45, DD55 e DD65 um coeficiente de variação maior do que o observado para as outras instâncias. Este comportamento novamente está ligado ao número de veículos em cada solução que compõem a média, ou seja, houve uma oscilação para cima na quantidade de veículos destas. Outra observação importante é que mesmo com uma frota mais restrita em termos da capacidade dos veículos, a única classe de instâncias onde houve fracionamento da coleta e entrega foi a BB, isso para instâncias com mais do que 30 requisições. O fracionamento limitou-se a não mais do que duas requisições por instância. Para todas as outras classes de instâncias não foi observado o fracionamento de coleta e entrega. Esperava-se uma redução no número de veículos usados, o que não foi observado, sendo este um dos benefícios mencionado por Nowak et al. (2008) e Archetti e Speranza (2012), ao se permitir fracionamento da coleta e entrega. Diante deste fato, uma futura investigação sobre a característica das instâncias usadas é válida. É perceptível que para as instâncias BB20, BB25, BB35, BB60, BB65 e CC15, os tempos levados para encontrar a melhor solução é menor do que a metade do tempo médio das soluções encontradas pela meta-heurística. Novamente, como mencionado na Seção 5.4, as hipóteses para tal comportamento são uma melhor solução inicial fornecida pela heurística construtiva e maior participação de uma ou mais heurísticas de remoção em relação às demais. Lembrando-se que a heurística construtiva seleciona requisições ao acaso para serem inseridas na melhor posição possível. Diante desses fatos, uma possível melhoria seria a criação de uma heurística construtiva com múltiplos reinícios e, assim, ao final seleciona-se a melhor solução inicial em termos do valor da função objetivo.

As Figuras 13, 14, 15 e 16 apresentam os resultados para as classes de instâncias AA, BB, CC e DD e têm interpretação semelhante daquelas da seção anterior. Observando-se as figuras é possível identificar que houve 11 instâncias com *Gaps* maiores que 1%. Destes, oito são referentes a instâncias da classe BB. Para as demais instâncias, todos os *Gaps* observados foram menores que 1%. Da mesma forma que feito na seção anterior, ao compararmos os *Gaps* aos

coeficientes de variação é possível identificar certos aspectos referentes a estabilidade da ALNS. A primeira constatação é que a classe de instância BB é a que possui mais aleatoriedade no processo de busca, o que pode ser devido ao método para realizar o fracionamento da coleta e entrega não ser muito eficiente. A segunda constatação é que, no geral, o processo de busca tem uma estabilidade aceitável passível de melhoria.

Uma comparação entre o modelo de três-índices e a meta-heurística, para as instâncias solucionadas pelo modelo, pode ser observada na Figura 17. Nesta última, representam-se as soluções obtidas pelo modelo, as melhores soluções encontradas pela meta-heurística, os tempos computacionais do modelo e da ALNS. De forma semelhante à Figura 12, a escala do eixo dos tempos teve o valor máximo fixado em 30 segundos, e somente os tempos referente ao modelo foram rotulados. Essas definições são estabelecidas para melhor visualização das informações.

Nota-se pela Figura 17 e juntamente pela análise da Tabela 10, que a meta-heurística proposta conseguiu encontrar soluções para todas as instâncias de todas as classes, com tempos computacionais menores do que os apresentados pelo modelo de três-índices. Observa-se também que para as instâncias resolvidas até a otimalidade pelo modelo, a ALNS encontrou soluções iguais em tempos computacionais menores. Ressalta-se também que para as instâncias AA20 e BB20, onde o modelo esgotou o tempo computacional para encontrar uma solução, a meta-heurística foi capaz de chegar a uma solução muito melhor em tempo computacional relativamente pequeno.

Por meio da análise dos testes computacionais do modelo de três-índices e da meta-heurística, e de posterior comparação entre as informações de ambos pela Figura 17, é possível constatar novamente que a formulação de três-índices proposta é mais adequada para instâncias pequenas de até 15 requisições, enquanto a meta-heurística proposta consegue lidar com instâncias de diferentes tamanhos, obtendo soluções factíveis dentro de tempos computacionais aceitáveis.

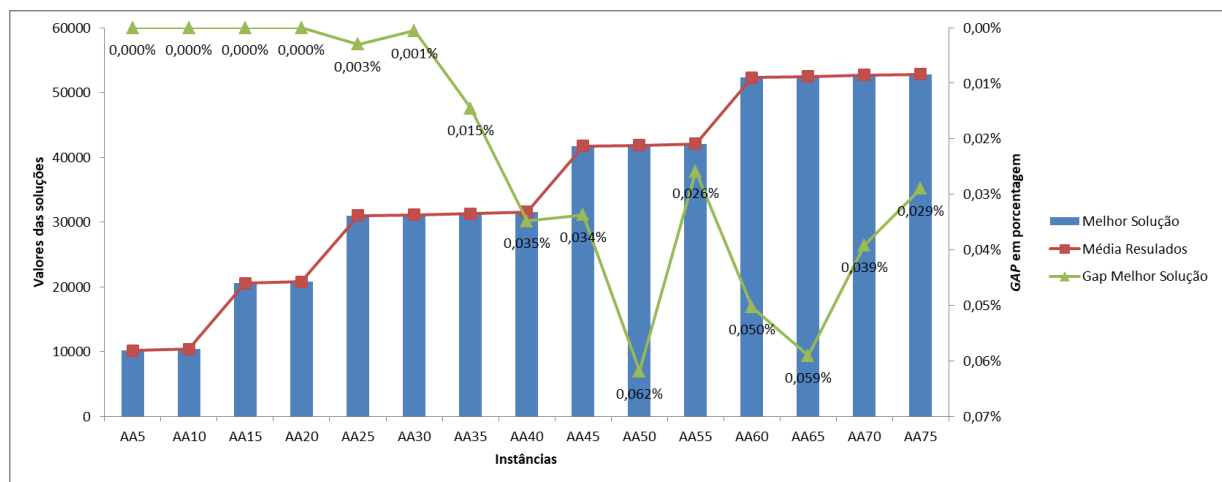


Figura 13: Soluções para a classe AA com coleta e entrega fracionada.

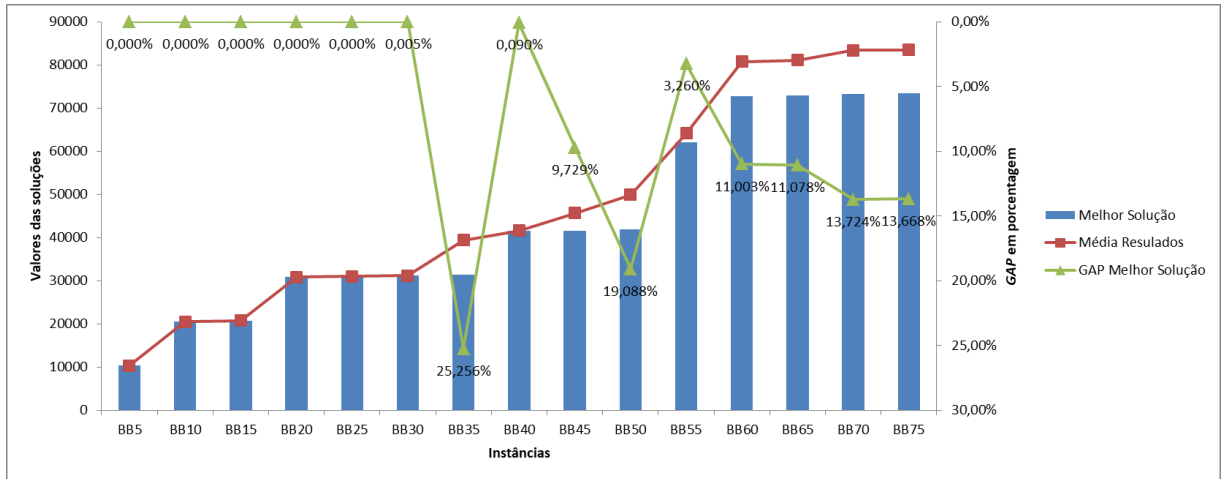


Figura 14: Soluções para a classe BB com coleta e entrega fracionada.

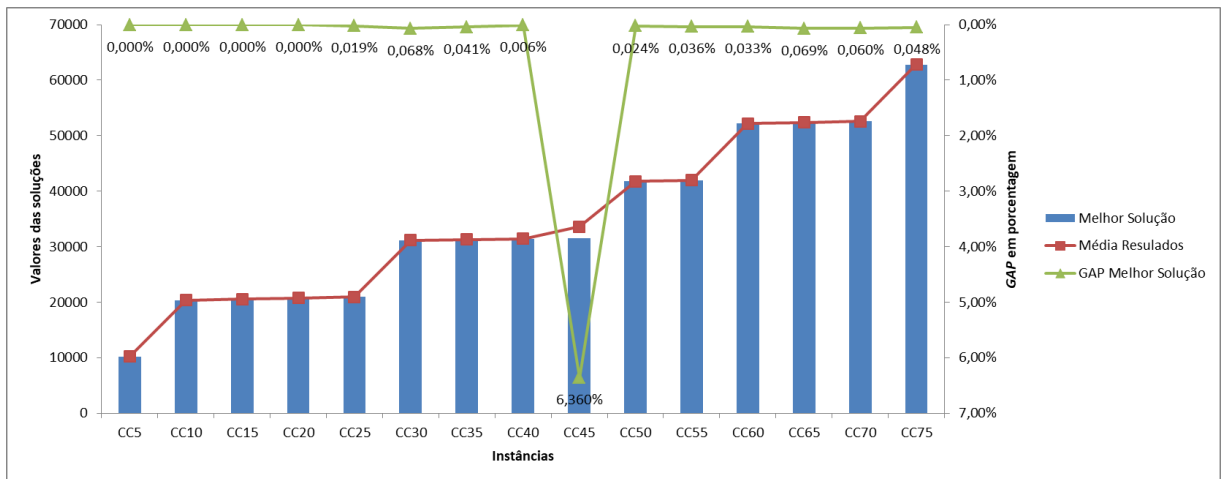


Figura 15: Soluções para a classe CC com coleta e entrega fracionada.

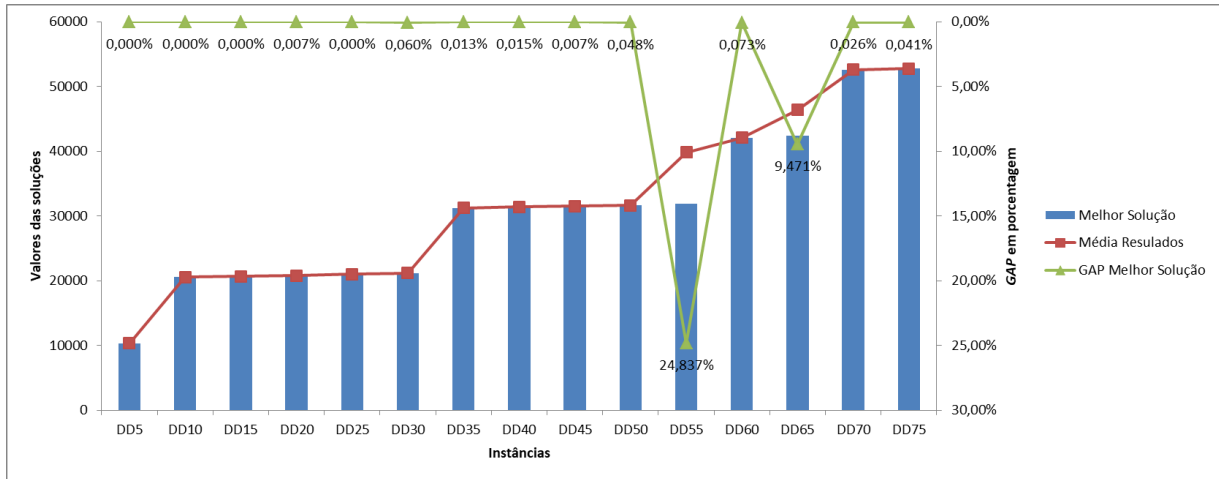


Figura 16: Soluções para a classe DD com coleta e entrega fracionada.

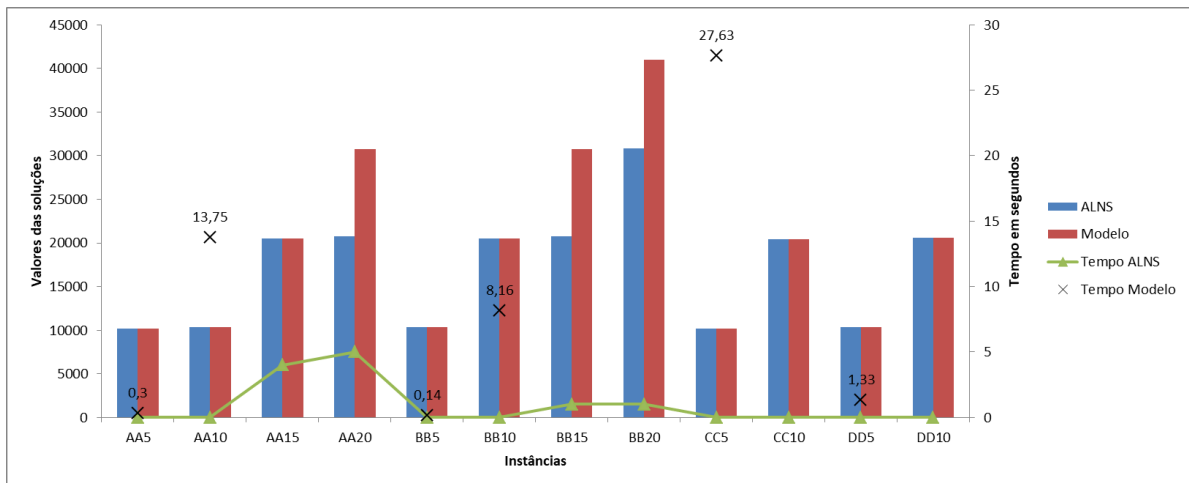


Figura 17: Comparação entre ALNS e modelo de três-índices.

Por fim, realizou-se um experimento onde a solução final da meta-heurística foi usada como solução inicial para o modelo de três índices. O número de veículos foi limitado no modelo de acordo com a quantidade de veículos usada na solução final da meta-heurística. Permitiu-se um tempo de 600 segundos e 25000 iterações para a ALNS obter uma solução. Logo em seguida o modelo de três-índices teve 3000 segundos para melhorar a solução inicial fornecida, totalizando em média 3600 segundos de tempo. Os resultados para o teste do modelo de três-índices com solução inicial provida pela ALNS estão descritos na Tabela 11. A primeira coluna apresenta as instâncias utilizadas, a segunda coluna contém as soluções fornecidas pela meta-heurística, a terceira coluna apresenta o respectivo tempo levado pela meta-heurística para encontrar a

solução reportada. Em seguida, para o modelo, a quarta coluna da tabela representa a respectiva solução obtida, a quinta coluna apresenta o tempo gasto pelo modelo para encontrar a solução reportada pelo CPLEX e, por fim, na última coluna é exibido o *Gap* da solução obtida. Caso o modelo esgote o tempo computacional para prover uma solução, reporta-se o tempo apenas como TL e caso o modelo falhe em obter uma solução dentro do tempo estabelecido é indicado pelo caracter “-”.

É possível observar pela Tabela 11 que o modelo conseguiu provar otimalidade para nove instâncias pequenas. Nota-se que o modelo só conseguiu melhorar uma única solução, a BB50, mas com um *Gap* ainda muito alto de 74%. Exceto para as instâncias pequenas, observa-se que as soluções ainda estão muito longe do ótimo com um *Gap* médio de 64%. Isso ressalta a ineficiência do modelo fazendo-se uso de um *software* de otimização de uso geral, pois o mesmo não foi capaz de melhorar a grande maioria das soluções providas inicialmente. Embora o modelo não tenha tido bom desempenho ao tentar melhorar as soluções fornecidas inicialmente pela ALNS, o mesmo proporcionou algumas informações pertinentes sobre as soluções encontradas pela meta-heurística. Nesse sentido, pode-se destacar que o modelo de três-índices comprovou que a ALNS encontrou soluções ótimas para algumas instâncias além de ter determinado um *Gap* para as demais. Nota-se também que a ALNS auxiliou o CPLEX a prover soluções, mesmo que com um *Gap* muito alto, que até então, não era possível sem uma solução inicial fornecida ao modelo, lembrando que o mesmo só foi capaz de reportar soluções para 12 instâncias dentre 60 da literatura.

Tabela 11: Resultados modelo de três-índices com solução inicial.

| Instância | Meta-Heurística | | Modelo de três-índices | | |
|-----------|-----------------|----------|------------------------|----------|--------|
| | Solução | Tempo(s) | Solução | Tempo(s) | Gap |
| AA5 | 10184,80 | 0,00 | 10184,80 | 0,08 | 0,00% |
| AA10 | 10383,60 | 0,00 | 10383,60 | 0,11 | 0,00% |
| AA15 | 20544,70 | 4,00 | 20544,70 | 70,58 | 2,00% |
| AA20 | 20764,30 | 4,00 | 20764,30 | TL | 0,00% |
| AA25 | 30986,70 | 138,00 | 30986,70 | TL | 34,00% |
| AA30 | 31121,90 | 184,00 | 31121,90 | TL | 62,00% |
| AA35 | 31302,60 | 203,00 | 31302,60 | TL | 66,00% |
| AA40 | 31551,40 | 416,00 | 31551,40 | TL | 66,00% |
| AA45 | 41699,90 | 592,00 | 41699,90 | TL | 74,00% |
| AA50 | 41910,10 | 591,00 | 41910,10 | TL | 98,00% |
| AA55 | 42119,30 | 508,00 | 42119,30 | TL | 98,00% |
| AA60 | 52347,10 | 593,00 | 52347,10 | TL | 98,00% |
| AA65 | 52490,90 | 341,00 | 52490,90 | TL | 98,00% |
| AA70 | 52694,40 | 432,00 | 52694,40 | TL | 98,00% |
| AA75 | 52838,00 | 545,00 | 52838,00 | TL | 98,00% |

continua

Tabela 11 – Resultados modelo de três-índices com solução inicial - continuação

| Instância | Meta-Heurística | | Modelo de três-índices | | |
|-----------|-----------------|----------|------------------------|----------|--------|
| | Solução | Tempo(s) | Solução | Tempo(s) | Gap |
| BB5 | 10339,70 | 0,00 | 10339,70 | 0,03 | 0,00% |
| BB10 | 20525,40 | 0,00 | 20525,40 | 0,38 | 0,00% |
| BB15 | 20737,30 | 0,00 | 20737,30 | 17,59 | 0,00% |
| BB20 | 30870,60 | 7,00 | 30870,60 | TL | 27,00% |
| BB25 | 31031,10 | 55,00 | 31031,10 | TL | 33,00% |
| BB30 | 31163,00 | 25,00 | 31163,00 | TL | 66,00% |
| BB35 | 41420,70 | 574,00 | 41420,70 | TL | 74,00% |
| BB40 | 41561,40 | 413,00 | 41561,40 | TL | 74,00% |
| BB45 | 51659,70 | 566,00 | 51659,70 | TL | 79,00% |
| BB50 | 41873,00 | 421,00 | 41872,70 | TL | 74,00% |
| BB55 | 72144,10 | 524,00 | 72144,10 | TL | 85,00% |
| BB60 | 62664,50 | 478,00 | 62664,50 | TL | 99,00% |
| BB65 | 83136,80 | 479,00 | 83136,80 | TL | 99,00% |
| BB70 | 83442,92 | 457,00 | 83442,92 | TL | 99,00% |
| BB75 | 83529,76 | 524,00 | 83529,76 | TL | 99,00% |
| CC5 | 10212,50 | 0,00 | 10212,50 | 0,13 | 0,00% |
| CC10 | 20385,50 | 0,00 | 20385,50 | TL | 49,00% |
| CC15 | 20571,20 | 11,00 | 20571,20 | TL | 49,00% |
| CC20 | 20764,40 | 10,00 | 20764,40 | TL | 97,00% |
| CC25 | 20961,70 | 20,00 | 20961,70 | TL | 97,00% |
| CC30 | 31128,30 | 99,00 | 31128,30 | TL | 98,00% |
| CC35 | 31281,20 | 201,00 | 31281,20 | TL | 98,00% |
| CC40 | 31431,20 | 391,00 | 31431,20 | TL | 98,00% |
| CC45 | 31561,80 | 92,00 | 31561,80 | TL | 98,00% |
| CC50 | 41792,00 | 597,00 | 41792,00 | TL | 99,00% |
| CC55 | 42017,90 | 481,00 | 42017,90 | TL | 99,00% |
| CC60 | 52185,50 | 547,00 | 52185,50 | TL | 99,00% |
| CC65 | 52402,80 | 270,00 | 52402,80 | TL | 99,00% |
| CC70 | 52645,60 | 473,00 | 52645,60 | TL | 99,00% |
| CC75 | 62820,20 | 364,00 | 62820,20 | TL | 99,00% |
| DD5 | 10324,80 | 0,00 | 10324,80 | 0,05 | 0,00% |
| DD10 | 20573,00 | 1,00 | 20573,00 | 468,14 | 7,00% |
| DD15 | 20680,30 | 1,00 | 20680,30 | TL | 49,00% |
| DD20 | 20779,80 | 47,00 | 20779,80 | TL | 49,00% |
| DD25 | 21009,30 | 15,00 | 21009,30 | TL | 97,00% |
| DD30 | 21150,40 | 78,00 | 21150,40 | TL | 97,00% |
| DD35 | 31246,80 | 207,00 | 31246,80 | TL | 98,00% |
| DD40 | 31384,80 | 264,00 | 31384,80 | TL | 98,00% |
| DD45 | 31510,70 | 364,00 | 31510,70 | TL | 98,00% |
| DD50 | 31640,50 | 304,00 | 31640,50 | TL | 98,00% |

continua

Tabela 11 – Resultados modelo de três-índices com solução inicial - continuação

| Instância | Meta-Heurística | | Modelo de três-índices | | |
|-----------|-----------------|----------|------------------------|----------|--------|
| | Solução | Tempo(s) | Solução | Tempo(s) | Gap |
| DD55 | 41822,90 | 596,00 | 41822,90 | TL | 99,00% |
| DD60 | 42031,80 | 547,00 | 42031,80 | TL | 99,00% |
| DD65 | 42359,70 | 333,00 | 42359,70 | TL | 99,00% |
| DD70 | 52595,50 | 302,00 | 52595,50 | TL | 99,00% |
| DD75 | 52793,60 | 584,00 | 52793,60 | TL | 99,00% |

No geral, a meta-heurística teve melhor desempenho frente aos modelos adaptados da literatura para o PCEJT-1 e PCJET-2. As soluções encontradas são aceitáveis dentro de tempos computacionais razoáveis.

5.6 Análise de Sensibilidade

A análise de sensibilidade foi conduzida de forma a se avaliar a contribuição que as heurísticas de remoção, bem como as heurísticas de inserção, têm no processo de busca da ALNS por melhores soluções. Primeiramente, analisaram-se as heurísticas de remoção de forma a se avaliar quais delas contribuem mais para a melhora das soluções. As heurísticas foram sendo habilitadas para uso uma a uma, até que todas pudessem ser usadas. Em seguida, analisaram-se as heurísticas de inserção, mais precisamente, a adição de ruído à função objetivo. Testes foram conduzidos com e sem adição de ruído à função objetivo e os resultados foram posteriormente comparados, de forma a se avaliar a necessidade de tal perturbação da solução. Todos os testes da análise de sensibilidade foram conduzidos para o PCEJT-2 devido a sua dificuldade de resolução por conter todas as restrições práticas abordadas neste trabalho. Sendo assim, as instâncias usadas foram as propostas por Ropke e Cordeau (2009) adaptadas para o PCEJT-2. Dentre estas instâncias, escolheu-se aquelas com 15, 40 e 65 requisições de coleta e entrega para as classes AA, BB, CC e DD. Cada instância foi executada cinco vezes usando-se a ALNS impondo-se um tempo de 600 segundos e 25000 iterações a ALNS.

Os resultados para a análise de sensibilidade podem ser vistos nas Tabelas 12 e 13. A interpretação de ambas é similar e descrita a seguir. Primeiramente, as instâncias utilizadas são apresentadas na primeira coluna. Em seguida, nas demais colunas, apresentam-se os valores médios da função objetivo (Tabela 12) e tempos (Tabela 13) para as cinco execuções para cada configuração (Config) da ALNS. Dessa forma, inicia-se a análise de sensibilidade somente com a heurística *Shaw removal* (S) habilitada e, em seguida, habilita-se uma heurística de remoção por vez, sendo elas: *Worst removal* (W), *Random removal* (R), *Time related removal* (T), *Distance related removal* (D), *Demand related removal* (De) e *Late arrival removal* (L). Cada

Config- representa uma quantidade de heurísticas de remoção habilitadas, a Config-1 possui somente a heurística de remoção S, a Config-2 contém S e W, já a Config-3 têm S, W e R, estão habilitadas na Config-4 as heurísticas S, W, R e T, para a Config-5 S, W, R, T e D, Config-6 têm S, W, R, T, D e De e por fim, a Config-7 possui todas as heurísticas de remoção habilitadas. Observa-se que as colunas das tabelas apresentam cada configuração obtida pela adição de uma nova heurística de remoção e estão dispostas nessa ordem respectivamente. A última coluna demonstra o coeficiente de variação dos valores da função objetivo ou tempos para os experimentos. Os valores em negrito salientam os menores valores médios para a função objetivo e tempo em cada instância.

Tabela 12: Resultados análise de sensibilidade - Valores médios da função objetivo

| Instâncias | Config-1 | Config-2 | Config-3 | Config-4 | Config-5 | Config-6 | Config-7 | Cof. var |
|------------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|----------|
| AA15 | 20544,67 | 20544,67 | 20544,67 | 20544,67 | 20544,67 | 20544,67 | 20544,67 | 0,0000 |
| BB15 | 20737,32 | 20737,32 | 20737,32 | 20737,32 | 20737,32 | 20737,32 | 20737,32 | 0,0000 |
| CC15 | 20571,25 | 20571,21 | 20571,18 | 20571,18 | 20571,18 | 20571,18 | 20571,18 | 0,0000 |
| DD15 | 20680,33 | 20680,33 | 20680,33 | 20680,33 | 20680,33 | 20680,33 | 20680,33 | 0,0000 |
| AA40 | 31568,83 | 31578,14 | 31557,06 | 31552,94 | 31556,01 | 31554,06 | 31555,58 | 0,0003 |
| BB40 | 41571,30 | 41604,80 | 41555,11 | 41577,36 | 41550,73 | 41569,29 | 41564,65 | 0,0004 |
| CC40 | 31428,89 | 31423,28 | 31421,25 | 31425,92 | 31420,62 | 31423,55 | 31422,10 | 0,0001 |
| DD40 | 31393,08 | 31393,15 | 31389,62 | 31387,27 | 31389,63 | 31390,15 | 31389,79 | 0,0001 |
| AA65 | 52511,91 | 52513,58 | 52503,97 | 52509,04 | 52482,98 | 52495,07 | 52493,46 | 0,0002 |
| BB65 | 81089,99 | 75055,86 | 83103,06 | 85362,27 | 81040,26 | 77041,21 | 77021,99 | 0,0464 |
| CC65 | 52446,00 | 52422,33 | 52424,30 | 52418,77 | 52436,66 | 52428,36 | 52400,02 | 0,0003 |
| DD65 | 44381,75 | 48399,89 | 46369,99 | 46364,56 | 46392,54 | 44354,28 | 48356,56 | 0,0353 |

Config-1: S; **Config-2:** S/W; **Config-3:** S/W/R; **Config-4:** S/W/R/T; **Config-5:** S/W/R/T/D; **Config-6:** S/W/R/T/D/De; **Config-7:** S/W/R/T/D/De/L

A Tabela 12 demonstra que para as instâncias AA15, BB15 e DD15, não foi notado melhora da função objetivo ao passo que aumenta-se o número de heurísticas de remoção. De fato, somente a heurística *Shaw removal* é suficiente para encontrar bons resultados. No entanto, para a instância CC15 o valor ótimo só foi encontrado quando três heurísticas de remoção estavam habilitadas. Nota-se para esta última instância, pouca diferença entre os valores médios da função objetivo encontrados por cada configuração da ALNS, dado que seu coeficiente de variação tende a zero.

Como pode ser visto na Tabela 12, as instâncias com 40 requisições demandaram quatro e cinco heurísticas de remoção habilitadas para os melhores resultados encontrados. O coeficiente de variação para estas instâncias foi próximo de zero, o que implica em uma baixa variação do valor da função objetivo entre as configurações testadas.

Ainda de acordo com a Tabela 12, a ALNS obteve melhor desempenho para instâncias AA65,

CC65 e DD65 com cinco, seis e sete heurísticas de remoção habilitadas. Esse comportamento não foi observado para a instância BB65, dado que, ao contrário das instâncias anteriores, a ALNS obteve melhor desempenho com somente duas heurísticas de remoção habilitadas. Para esta instância, valores próximos da função objetivo encontrados pela Config-2 foram obtidos por Config-6 e -7, e outras configurações não obtiveram bons resultados. Diante desse fato, é possível supor que se houvesse mais tempo para ALNS sob as configurações Config-6 e -7, soluções iguais ou até melhores que a encontrada pela meta-heurística sob a Config-2, poderiam ser achadas.

A seguir, apresenta-se a Tabela 13 que tem o intuito de verificar a variação dos tempos levados pela ALNS para encontrar uma solução de qualidade aceitável durante o processo de busca, a medida que aumenta-se o número de heurísticas de remoção habilitadas a operar. A interpretação das colunas é a mesma da Tabela 12, atenta-se somente para os tempos marcados em negrito que evidenciam a configuração onde foi encontrada a menor solução em termos da função objetivo. Os menores tempos são destacados por asterisco (*) e não necessariamente coincidem com as menores soluções encontradas.

Tabela 13: Resultado análise de sensibilidade - Tempos médios para às soluções

| Instâncias | Config-1 | Config-2 | Config-3 | Config-4 | Config-5 | Config-6 | Config-7 | Cof. var |
|------------|----------|----------------|----------|----------|---------------|----------------|---------------|----------|
| AA15 | 9,60 | 3,60 | 2,60 | *2,00 | 2,40 | 2,60 | 2,60 | 0,7377 |
| BB15 | 2,20 | 0,40 | 0,60 | *0,20 | 0,80 | 0,60 | 0,40 | 0,9025 |
| CC15 | 18,80 | 12,40 | 8,60 | 6,40 | *2,60 | 3,80 | 6,00 | 0,6700 |
| DD15 | 1,60 | 2,00 | 3,00 | *1,00 | 1,60 | 1,40 | 1,20 | 0,3926 |
| AA40 | 349,00 | 299,00 | 373,00 | 178,40 | 222,80 | 282,20 | *177,40 | 0,2922 |
| BB40 | 469,40 | *355,60 | 416,60 | 364,60 | 472,80 | 403,40 | 386,20 | 0,1142 |
| CC40 | 400,40 | 481,20 | 440,40 | 445,60 | 278,00 | *247,80 | 309,20 | 0,2480 |
| DD40 | 478,60 | 422,40 | 442,00 | 454,40 | 352,40 | 414,60 | *280,80 | 0,1675 |
| AA65 | 394,60 | 419,80 | 380,40 | 327,20 | 326,40 | 407,00 | *318,00 | 0,1163 |
| BB65 | 384,00 | *235,20 | 337,00 | 242,40 | 322,80 | 373,00 | 478,40 | 0,2497 |
| CC65 | 447,20 | 438,00 | 300,00 | 385,80 | 346,00 | *253,60 | 365,40 | 0,1934 |
| DD65 | 431,80 | *401,80 | 422,60 | 424,20 | 501,00 | 450,20 | 484,60 | 0,0805 |

Config-1: S; Config-2: S/W; Config-3: S/W/R; Config-4: S/W/R/T; Config-5: S/W/R/T/D; Config-6: S/W/R/T/D/De; Config-7: S/W/R/T/D/De/L

Pela Tabela 13 é possível identificar que para as instâncias AA15, BB15 e CC15, o tempo computacional foi diminuindo a partir da habilitação de mais heurísticas de remoção. Esse decréscimo foi observado até a Config-4 para a instância AA15 e até a Config-5 para as instâncias BB15 e CC15. A partir das configurações mencionadas anteriormente, o tempo computacional foi aumentando para as referidas instâncias. Para a instância DD15, observou-se um comportamento diferente, onde o tempo computacional foi aumentando até a Config-3, tendo uma queda sob a Config-4, com aumento na Config-5 e posterior declínio do tempo computacional nas

posteriores configurações. No entanto, apesar das particularidades do movimento dos tempos para cada instância, é possível observar que todas as instâncias demonstram uma tendência de queda dos tempos computacionais, umas com grau mais acentuado outras não. Em relação ao coeficiente de variação, a instância com 15 requisições que mais apresentou variação em seus tempos foi a BB15, além disso, esse conjunto de instâncias com 15 requisições dentre os outros dois conjuntos com 40 e 65 requisições, foi o que mais apresentou variação em seus tempos. O movimento dos tempos computacionais pode ser melhor compreendido pela Figura 18.

De acordo com a Tabela 13, é identificável que para a instância AA40 os tempos se mantêm quase os mesmos entre a Config-1 e Config-3, com uma queda dos mesmos para a Config-4, seguida de um aumento dos tempos para a Config-5 e -6 terminando com decréscimo do tempo para a Config-7. Nota-se que, para instância BB40 uma queda do tempo para a Config-2 onde o mesmo tem pouca variação até a Config-4, o tempo volta a aumentar para a Config-5 seguindo uma queda para as demais Config-6 e -7. Identifica-se também que inicialmente para a instância CC40 o tempo tem ligeiro aumento na Config-2, mantendo-se quase o mesmo até a Config-4 que a partir desta configuração, observa-se declínio do mesmo para a Config-5 e -6 seguido de outro pequeno aumento para a Config-7. Por final, tem-se que o tempo para a instância DD40 sofre uma pequena queda para a Config-2 mantendo-se quase o mesmo até a Config-4 sendo que, na Config-5, observa-se uma queda seguida de um aumento para a Config-6 terminando com declínio do tempo para a Config-7. Em suma, é possível identificar que para as instâncias AA40, CC40 e DD40 existe uma tendência de queda dos tempos computacionais, embora para a instância BB40 haja uma tendência de os tempos não oscilarem muito. Os tempos para o conjunto de instâncias com 40 requisições podem ser visualizados na Figura 19.

É possível observar oscilações dos tempos computacionais praticados pela ALNS, à medida que se aumenta o número de heurísticas de remoção habilitadas. Para a instância AA65, a partir da Config-2, nota-se uma tendência de queda dos tempos até a Config-5. Em seguida, os tempos oscilam entre aumento e queda para a Config-6 e -7, respectivamente. A instância BB65 tem oscilações nos tempos até a Config-4, sendo que a partir desta os tempos começam a aumentar. Para a instância CC65, observa-se um intervalo de queda compreendido entre a Config-1 e Config-3, posteriormente é visível o aumento do tempo para a Config-4 com decréscimo do mesmo para as Config-5 e -6, tendo por fim, um aumento novamente para a Config-7. Por último, temos a instância DD65 que apresenta uma tendência de aumento dos tempos com pequenas oscilações para menos para a Config-2 e Config-6. Em resumo, as instâncias AA65 e CC65 apresentam tendência de queda dos tempos computacionais à medida que aumenta-se o número de heurísticas de remoção. Enquanto que as instâncias BB65 e DD65 apresentam tendência de aumento dos tempos computacionais conforme se aumenta quantidade de heurísticas de remoção. Estas oscilações podem ser melhor observadas na Figura 20.

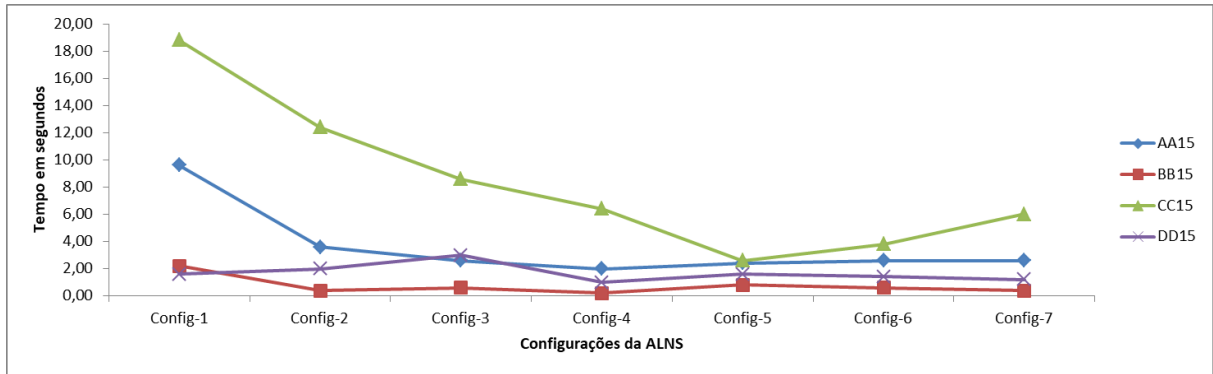


Figura 18: Análise de sensibilidade dos tempos para instâncias com 15 requisições.

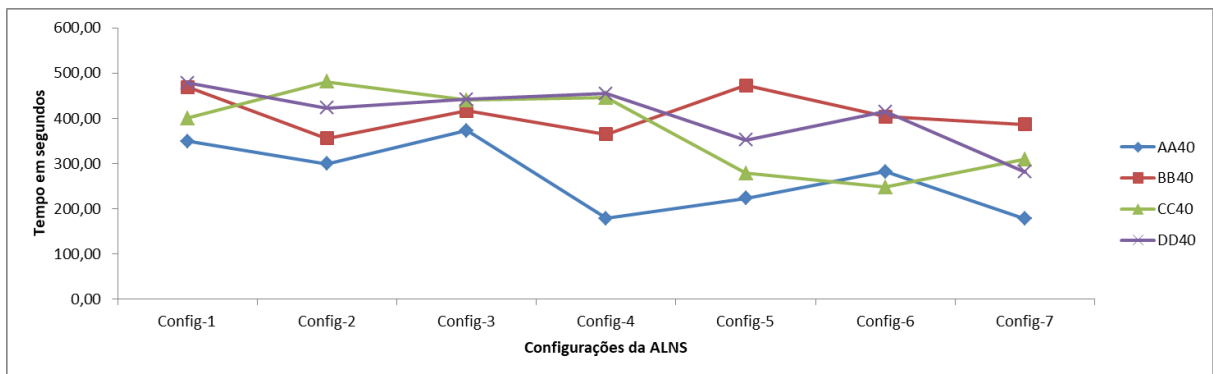


Figura 19: Análise de sensibilidade dos tempos para instâncias com 40 requisições.

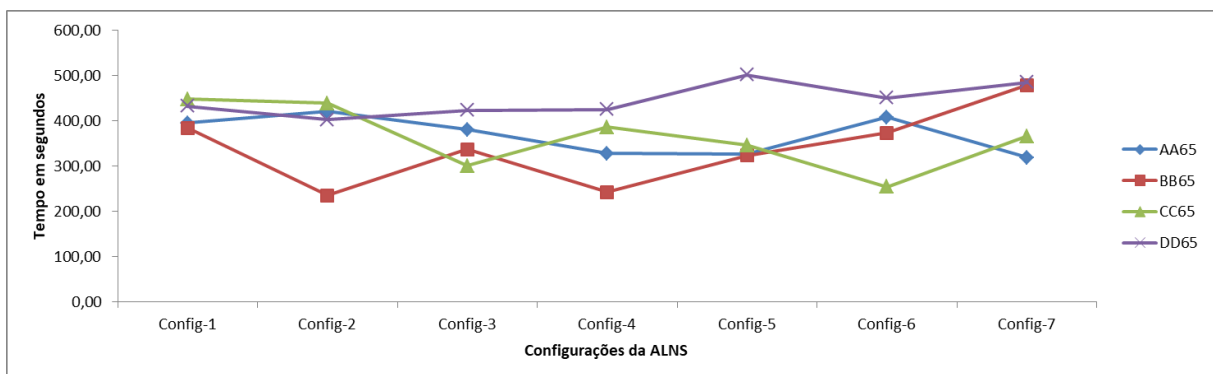


Figura 20: Análise de sensibilidade dos tempos para instâncias com 65 requisições.

A Tabela 14 sumariza a melhor configuração encontrada para cada instância. Caso os valores da função objetivo para as soluções sejam iguais, reporta-se a configuração que obteve o menor

tempo. A primeira coluna mostra as instâncias usadas, a segunda coluna apresenta o valor da função objetivo para a melhor solução encontrada e a terceira coluna apresenta com qual configuração a meta-heurística conseguiu encontrar a melhor solução.

Dessa forma, analisando a Tabela 14 juntamente com a Tabela 12, é visível que praticamente todas as configurações encontraram a melhor solução para instâncias com 15 requisições. Desse modo, avalia-se que nem todas as heurísticas de remoção estão ajudando o processo de busca. Atentando para estes resultados, aparentemente cinco tipos de vizinhanças exploradas por meio das heurísticas de remoção são mais que suficientes para encontrar boas soluções em tempos computacionais pequenos, isso referindo-se a instâncias pequenas com até 15 requisições.

Para as instâncias com 40 requisições, de acordo com a Tabela 14 e Tabela 12, observa-se que boas soluções foram encontradas com quatro heurísticas de remoção habilitadas para as instâncias AA40 e DD40. Já para as instâncias BB40 e CC40, soluções com boa qualidade foram obtidas com cinco heurísticas de remoção atuando no processo de busca.

Para as instâncias AA65, BB65, CC65 e DD65, foi observado que o número de heurísticas de remoção habilitadas que proporcionou boas soluções foram cinco, duas, sete e seis heurísticas, respectivamente. Exceto pela instância BB65, identifica-se que a partir do momento que as instâncias se tornam maiores e mais difíceis, maior foi o quantidade de heurísticas de remoção usadas. No caso da instância BB65, onde somente duas heurísticas foram necessárias para se encontrar soluções de boa qualidade, pode-se considerar algumas hipóteses. A primeira seria uma solução inicial de melhor qualidade provida pela heurística construtiva. A segunda seria que a estrutura dessa instância em particular foi melhor explorada pela Config-2. Por fim, a terceira hipótese seria a própria aleatoriedade que ocasionou que uma boa solução fosse encontrada somente com duas heurísticas. Mais experimentos teriam que ser conduzidos para verificar as hipóteses mencionadas.

No geral, apesar de ser possível identificar uma tendência de aumento do número de heurísticas de remoção atuantes no processo de busca, ao passo que as instâncias se tornam maiores e mais difíceis, não é possível afirmar que quanto maior a quantidade de heurísticas de remoção melhor serão as chances de se encontrar soluções de boa qualidade para problemas maiores. Na verdade, estudos como o Christiaens e Vanden Berghe (2018) apontam que um número menor de heurísticas, que sejam melhor projetadas, seriam mais eficientes frente ao aumento sem critério das mesmas. Outro ponto que deve ser considerado é a própria estrutura de dados dos mecanismos de remoção, que pode interferir tanto para melhora quanto para piora do desempenho da ALNS. No entanto, mais testes teriam que ser conduzidos para avaliar essa interferência.

A Tabela 15 apresenta os resultados da análise de sensibilidade para as heurísticas de inserção com e sem ruído. A primeira coluna da tabela apresenta as instâncias utilizadas, a segunda e terceira colunas apresentam a média dos valores da função objetivo das soluções (MS-) e seus

Tabela 14: Melhor Configuração por instância

| Instâncias | Melhor Solução | Config. | Melhor Solução |
|------------|----------------|---------|----------------|
| AA15 | 20544,67 | | Config-4 |
| BB15 | 20737,32 | | Config-4 |
| CC15 | 20571,18 | | Config-5 |
| DD15 | 20680,33 | | Config-4 |
| AA40 | 31552,94 | | Config-4 |
| BB40 | 41550,73 | | Config-5 |
| CC40 | 31420,62 | | Config-5 |
| DD40 | 31387,27 | | Config-4 |
| AA65 | 52482,98 | | Config-5 |
| BB65 | 75055,86 | | Config-2 |
| CC65 | 52400,02 | | Config-7 |
| DD65 | 44354,28 | | Config-6 |

Config-1: S; **Config-2:** S/W; **Config-3:** S/W/R; **Config-4:** S/W/R/T; **Config-5:** S/W/R/T/D; **Config-6:** S/W/R/T/D/De; **Config-7:** S/W/R/T/D/De/L

respectivos tempos médios computacionais (MT-) para os testes sem ruído. A terceira e quarta colunas apresentam a média dos valores da função objetivo das soluções (MS+) e os tempos computacionais médios (MT+) para os testes com ruído. As duas últimas colunas apresentam o *Gap* dos valores médios (função objetivo e tempos) entre os testes sem ruído e com ruído. O *Gap* é calculado pela expressão (5.4), onde $V_{ruído}^{(-)}$ são os valores dos teste sem ruído (funções objetivos ou tempos) e $V_{ruído}^{(+)}$ são os valores dos testes com ruído.

$$Gap = \left\{ \frac{V_{ruído}^{(-)} - V_{ruído}^{(+)}}{V_{ruído}^{(+)}} \right\} .100. \quad (5.4)$$

Ao analisar a Tabela 15, é possível identificar que a ALNS operando sem a adição de ruído encontrou melhores soluções para cinco instâncias AA40, AA65, BB40, BB65, DD65 de 12 utilizadas nos testes, melhorando as soluções em 0,030%, 0,008%, 0,005%, 7,297% e 8,653%. Para as instâncias CC40, CC65, DD40, a ALNS com adição de ruído, encontrou respectivamente, resultados 0,034%, 0,002% e 0,003% melhores em comparação com os resultados sem adição de ruído. Nas demais instâncias, o desempenho referente ao valor das funções objetivos foi semelhante, com *Gap* zero. Em relação aos experimentos sem adição de ruído para as instâncias onde encontrou-se melhores resultados, também reporta-se melhora dos tempo computacionais para as instâncias AA40, AA65 e DD65. Para as instâncias onde os experimentos com adição de ruído tiveram melhor desempenho, observam-se tempos melhores para as instâncias DD40 e CC65. A adição de ruído proporcionou menores tempos computacionais para as instâncias AA15 e CC15, enquanto que a ALNS operando sem ruído obteve menores tempos computacionais para as instâncias BB15 e DD15.

Tabela 15: Análise de sensibilidade para as heurísticas de inserção com ruído e sem ruído

| Instâncias | MS- | MT- | MS+ | MT+ | Gap S+ | Gap T+ |
|------------|----------|--------|----------|--------|---------|----------|
| AA15 | 20544,67 | 3,80 | 20544,67 | 2,00 | 0,000% | 90,000% |
| BB15 | 20737,32 | 0,20 | 20737,32 | 0,60 | 0,000% | -66,667% |
| CC15 | 20571,21 | 9,80 | 20571,18 | 2,60 | 0,000% | 276,923% |
| DD15 | 20680,33 | 1,20 | 20680,33 | 1,80 | 0,000% | -33,333% |
| AA40 | 31552,92 | 179,00 | 31562,34 | 280,60 | -0,030% | -36,208% |
| BB40 | 41548,20 | 453,60 | 41550,23 | 321,40 | -0,005% | 41,133% |
| CC40 | 31428,00 | 178,40 | 31417,45 | 310,40 | 0,034% | -42,526% |
| DD40 | 31388,23 | 416,20 | 31387,26 | 372,60 | 0,003% | 11,702% |
| AA65 | 52473,20 | 490,20 | 52477,20 | 492,40 | -0,008% | -0,447% |
| BB65 | 75143,10 | 363,80 | 81057,89 | 320,40 | -7,297% | 13,546% |
| CC65 | 52402,18 | 484,00 | 52400,97 | 369,00 | 0,002% | 31,165% |
| DD65 | 42340,96 | 459,60 | 46351,74 | 531,40 | -8,653% | -13,511% |

Com base nos testes conduzidos para as heurísticas de inserção, pela posterior análise de seus resultados e com os valores atuais dos parâmetros da ALNS, observa-se que, aparentemente a meta-heurística apresentou melhores resultados sem a adição de ruído na função objetivo para cinco instâncias enquanto que a adição ruído proporcionou melhores resultados para três instâncias, isso para o conjunto de instâncias utilizado nesta análise de sensibilidade. Embora os testes sem adição de ruído obtiveram um desempenho melhor, somente para dois casos houve melhoras significativa em torno de 7 a 8%. Estes resultados são sumarizados na Figura 21.

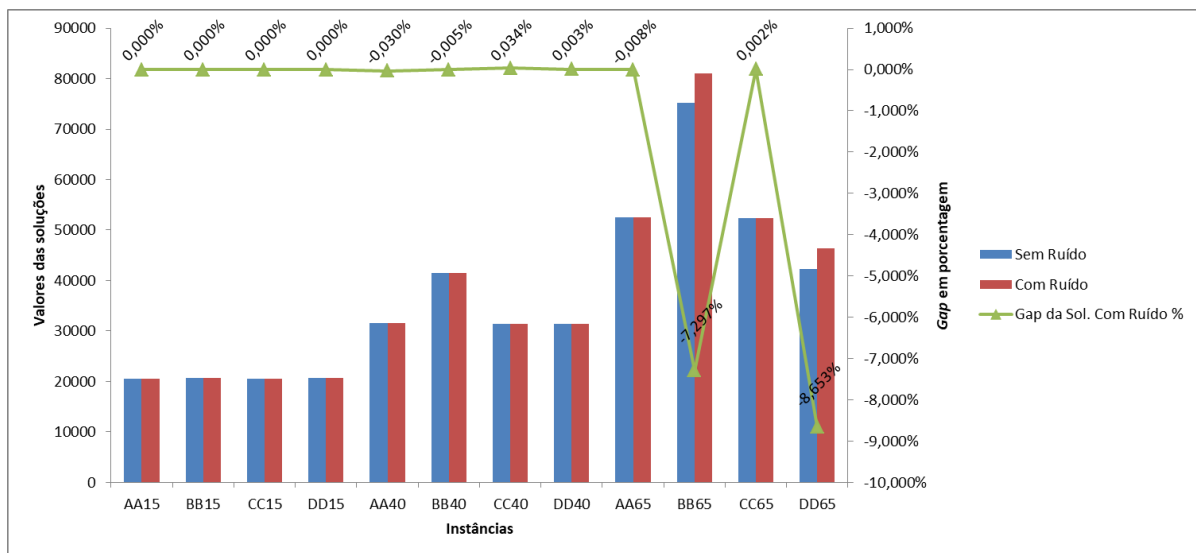


Figura 21: Análise de sensibilidade das heurísticas de inserção com e sem ruído.

Em síntese, os experimentos de análise de sensibilidade para as heurísticas de remoção evidenciaram um possível comportamento da meta-heurística quanto ao tamanho das instâncias. Observou-se que quanto maiores às instâncias mais vizinhas são exploradas para obter-se soluções de qualidade. No entanto, não é possível afirmar conclusivamente que esse é um comportamento padrão da ALNS. Deve-se lembrar de que as heurísticas de remoção foram sendo adicionadas uma a uma e não foram realizados experimentos com todas as combinações possíveis entre elas. Pode haver combinações entre as heurísticas de remoção que sejam melhores para um conjunto determinado de instâncias, mas que não o sejam para outros. Para os experimentos das heurísticas de inserção com e sem adição de ruído, observou-se que a não adição deste, melhorou algumas soluções. Esse comportamento não era esperado, pois esperava-se que o ruído auxiliasse a ALNS a escapar de ótimos locais, melhorando a solução incumbente e não piorando-a (Ropke e Pisinger, 2006). No entanto, sem mais experimentos em instâncias maiores, até mesmo reais e com outros valores para o parâmetro de ruído, não é possível atestar que a adição de ruído seja completamente desnecessária.

Capítulo 6

Software para web

A ferramenta computacional apresentada neste capítulo consiste em um *software* voltado para o ambiente *web*. Esse sistema tem o objetivo de proporcionar uma maneira fácil e clara para o uso dos modelos e métodos desenvolvidos neste trabalho, ocultando do usuário final a complexidade inerente ao uso dos modelos e métodos propostos e provendo apoio à tomada de decisão. Na verdade, a interface desenvolvida por esta pesquisa visa ir além de facilitar somente o uso dos modelos e métodos. Apresenta-se uma interface gráfica *web open-source* para o usuário final, disponível publicamente e capaz de ser estendida e utilizada para mediar e facilitar o uso de modelos e métodos referentes a diferentes variantes do Problema de Roteamento de Veículos. Para tal, *softwares*, bibliotecas e outras ferramentas computacionais livres para uso, e/ou *open-source*, foram empregadas no seu desenvolvimento. É necessário salientar que o processo de engenharia de *software* envolvido na confecção do sistema supracitado não foi trivial. Isso foi devido à integração de tecnologias diferentes e também ao fato de ser uma ferramenta que almeja ser reutilizável, o que de certa forma, coloca grande esforço em conceber-se uma arquitetura capaz de lidar com tipos diferentes de PRVs, tanto em nível de banco de dados como em nível de linguagem de programação.

6.1 Tecnologia para o desenvolvimento

Utilizou-se Java como linguagem de programação, pois permite a portabilidade do código entre diferentes sistemas operacionais e também por seu uso, e da maioria de suas bibliotecas desenvolvidas por terceiros, ser gratuito. Como servidor de banco de dados relacional usa-se o MySQL 5.7 *Community*, escolhido por seu uso bem difundido entre desenvolvedores, além de apresentar a robustez necessária para armazenar e manipular os dados posteriormente gravados. Como *framework* para mapear objetos para bancos de dados relacionais utilizou-se o Hibernate 5.2.13 juntamente com a biblioteca de persistência JPA (do inglês *Java Persistence API*).

Devido ao fato de o sistema ter sido desenvolvido em Java e para o ambiente *web*, é necessário o emprego de um servidor que possibilite o uso do *software* por meio de um navegador de Internet. Dessa forma, o servidor Apache TomCat 9.0 foi usado para esta finalidade. Sua escolha foi pautada por ser um *software open-source*, pela robustez e pela grande comunidade envolvida, garantindo a sua perpetuação e manutenção, além de ser útil no caso de dúvidas, possíveis erros ou dificuldades que venham a ser encontradas futuramente.

A comunicação com o código desenvolvido em C++ (modelos e metaheurística) é feita por meio da biblioteca JNA 4.5.2 (do inglês *Java Native Access*). Ela é responsável por comunicar-se com bibliotecas implementadas em código nativo como C e C++, usando apenas linguagem de programação Java, sendo esta a motivação pela sua escolha.

Para que o sistema em C++ possa operar é necessário, dentre outros parâmetros, enviar a ele as matrizes de distâncias e tempos entre os nós (clientes) do problema. Essas matrizes são obtidas por meio de mapas reais providos pelo *OpenStreetMap* (OSM), que é um serviço *open source* de fornecimento de dados geográficos havendo liberdade para uso dos mesmos. Após esta etapa de obtenção dos mapas, estes devem ser tratados por um servidor de roteamento *open source*, desenvolvido em C++ por Luxen e Vetter (2011) denominado OSRM (do inglês *Open Source Routing Machine*). Ele é responsável por calcular o caminho mais curto em redes de ruas e estradas retornando as matrizes de distâncias e tempos necessárias. É importante mencionar que cada região (ou seja, um mapa) obtida pelo OSM deve ser previamente pré-processado pelo OSRM. Esse pré-processamento requer tempo e ele é feito para gerar informações sobre tempos de deslocamento, entre pontos, dado pelo caminho mínimo entre eles. Quanto maior a região geográfica do mapa, mais tempo leva-se para essa etapa. No entanto, após esse passo, as matrizes mencionadas anteriormente são obtidas com certa rapidez. O pré-processamento é feito uma única vez para qualquer mapa e as informações podem ser usadas por tempo indeterminado, em qualquer aplicação e fora do escopo do sistema *web* desenvolvido neste trabalho.

A comunicação entre o OSRM e a interface gráfica desenvolvida em Java é feita pelo protocolo HTTP. A biblioteca *Apache HttpComponents* foi escolhida para lidar com essa comunicação. Ela nada mais é do que um conjunto de ferramentas para construir serviços clientes e servidores para o protocolo HTTP.

Para a interface gráfica, utiliza-se a biblioteca padrão de componentes *web* Apache MyFaces 2.2.12 como implementação da especificação Java Server Faces (JSF) e a biblioteca de componentes ricos para JSF denominada Primefaces 6.2. Para a interação com mapas oferecidos pelo OSM, utiliza-se a biblioteca *open source Leaflet* 1.3.4 feita em JavaScript. Essa interação vai além de marcar pontos nos mapas, permitindo também a busca por endereço por meio de localização geográfica (latitude e longitude) ou o inverso, isto é, busca pela localização geográfica por meio do endereço. As buscas são feitas por meio de um serviço oferecido pelo OSM denominado *Nominatim*. Cabe também dizer que o formato de troca de dados JSON (do inglês *JavaScript Object Notation*) é usado nas respostas do servidor OSRM e do serviço *Nominatim* do OSM, sendo a biblioteca JSON-java (org.json) usada para decodificá-las. Por fim, como ambiente de desenvolvimento integrado (IDE) usou-se o Eclipse Oxygen.

6.2 Documentação e funcionamento do *software*

A documentação do sistema *web* é composta por *JavaDocs*, diagramas de casos de uso, diagramas de classes, diagrama EER (do inglês *Extended Entity-Relationship*) de banco de dados e das interfaces gráficas. *JavaDocs* são descrições detalhadas do funcionamento das classes e dos métodos e atributos pertencentes a elas. O diagrama de casos de uso é estabelecido para mapear as possíveis interações que os atores venham a ter com o *software*. Os atores podem ser usuários finais ou outros sistemas. Já o diagrama de classes apresenta uma abstração dos objetos que compõem um sistema e seus relacionamentos, operações e atributos. Os diagramas são importantes para a modelagem e o desenvolvimento do sistema e ainda mais necessários para futuras manutenções e alterações nos códigos e funcionalidades (Pressman, 2009; Booch et al., 2006).

6.2.1 Diagramas de casos de uso, classes e EER

O diagrama de casos de uso da Figura 22 representa a interação típica que o usuário pode ter com o sistema *web* e com o sistema em C++. Dessa forma, o usuário é capaz de gerenciar clientes, veículos e produtos. Esse gerenciamento dá-se pela inclusão, alteração, exclusão e consulta ao banco de dados. Também é possível ao usuário gerenciar o plano de roteamento, ou seja, a interação que este terá com a interface gráfica. Essa interação permite realizar operações de criação de requisições de coleta e entrega, vinculações das demandas dos produtos aos clientes das requisições, seleção de quais veículos irão compor a frota do problema e, por final, possibilita ao usuário resolver o problema e visualizar as rotas criadas. Os relacionamentos entre casos de uso marcados como *include*, determinam a dependência entre eles. Percebe-se então que é necessário que o sistema C++ devolva uma solução do problema para o sistema desenvolvido em Java ou informe sobre a não resolução do mesmo, de forma obrigatória.

Concernente aos diagramas de classes, optou-se por apresentar a modelagem das classes do *software web* separadamente por razões de clareza do modelo e entendimento do mesmo. Dessa forma, dividiram-se os diagramas de classes em quatro, de modo que o primeiro refere-se às classes que modelam um cliente, o segundo apresenta aquelas que modelam o plano de roteamento, um terceiro diagrama demonstra as classes usadas para modelar um problema de PCEJT e, por fim, o quarto diagrama contém as classes usadas para modelar a comunicação com o sistema em C++ que contém a ALNS. A seguir, apresenta-se sucintamente a descrição de cada diagrama.

O diagrama de classes do cliente é visto na Figura 23. Neste diagrama um cliente é representado pela classe *Client*, que por sua vez deve estar vinculado a um tipo *ClientType*, sendo que, um cliente representa um nó ou um depósito do problema real. Um cliente está associado a uma localização, representada pela classe *Location*, que é uma abstração do endereço geo-referenciado.

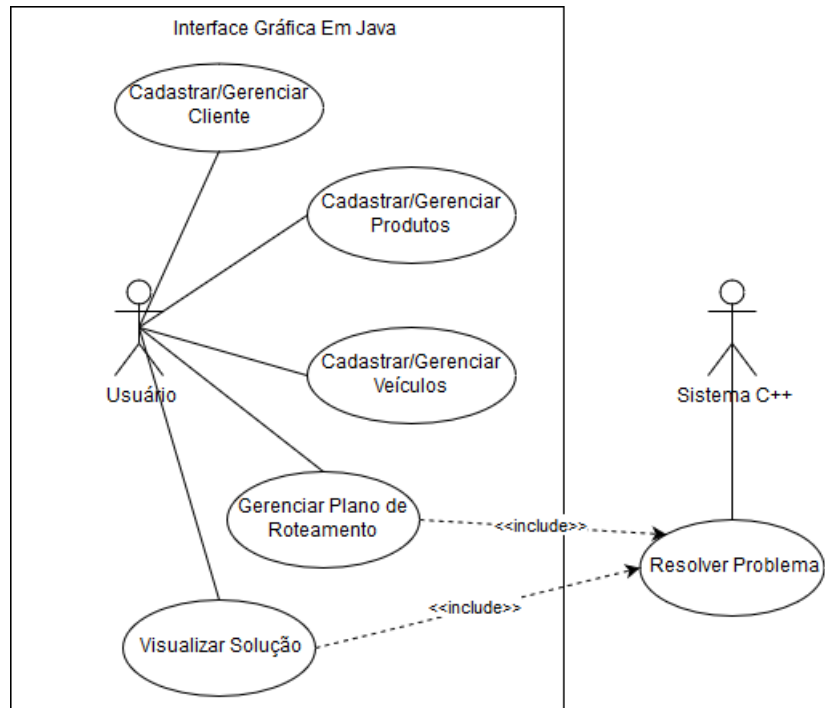


Figura 22: Casos de uso.

A classe *Demand* modela uma demanda de um cliente que por sua vez está associada a um único produto modelado pela classe *Product*. Tanto produtos como demandas possuem uma unidade de medida, abstraída por uma enumeração *UnitMeasure*. Esse tipo de modelagem é necessário pois um produto pode ser armazenado em determinada unidade de medida e posteriormente usado ou transportado em outra. Cada cliente pode demandar vários produtos, dessa forma existe uma lista de demandas atribuída a cada cliente. A classe *ClientTime* captura a restrição de janelas de tempo. No caso, optou-se por permitir que um cliente possa ter múltiplas janelas de tempo. Embora essa não seja uma variante abordada neste trabalho, futuras extensões do *software web* para esse caso particular não necessitarão de grandes alterações nos códigos. Por último, as restrições de paridade e precedência de visita aos clientes são abstraídas e modeladas pela classe *Pairing*.

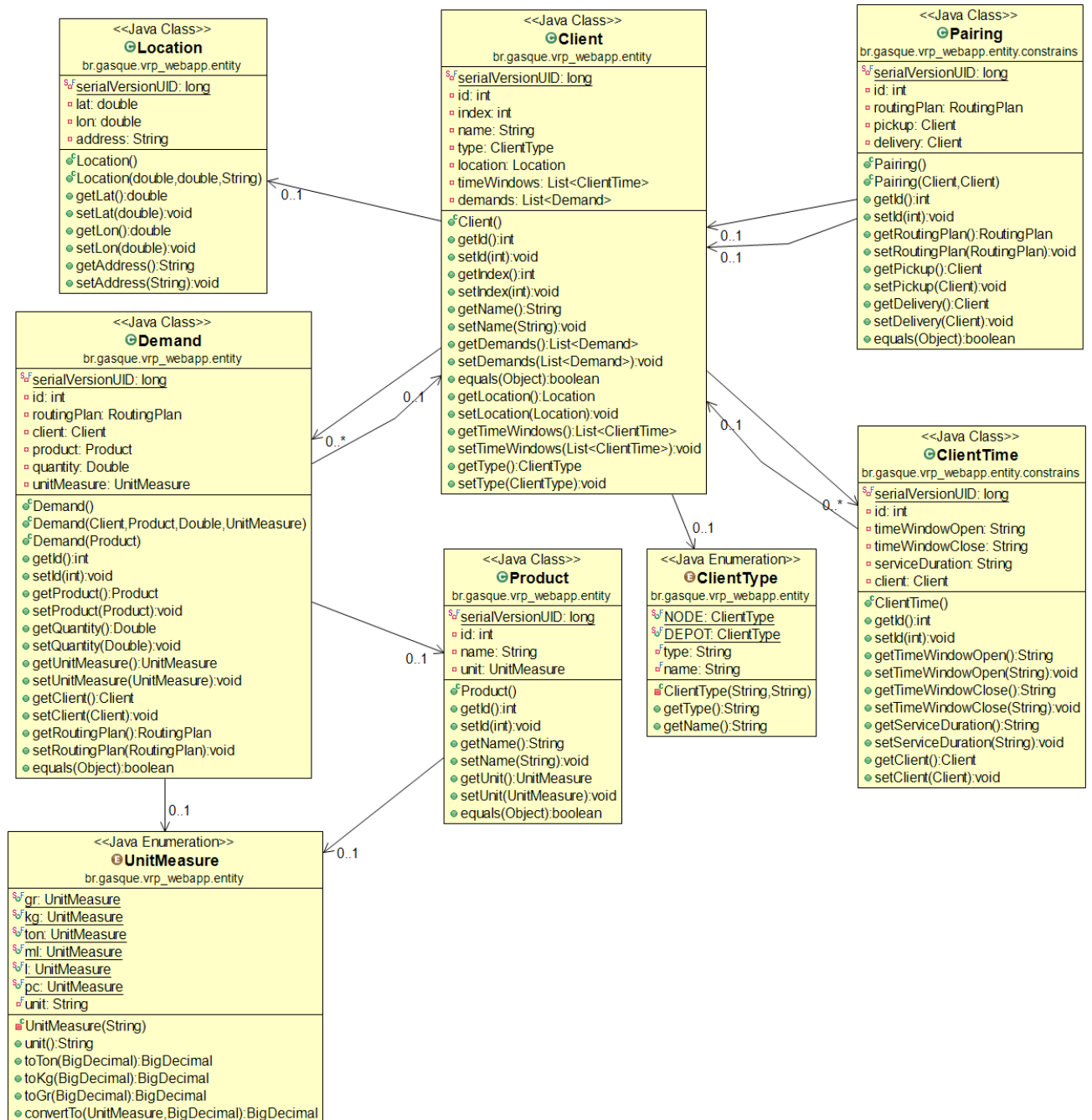


Figura 23: Diagrama de classes de cliente

Observe que, para a modelagem de um plano de roteamento a ser solucionado, somente são necessárias as classes *RoutingPlan*, que é a representação do plano de roteamento, e outras duas, *Demand* e *Client*. O plano de roteamento possui uma lista de rotas, que são representadas pela classe *Route*, que por sua vez tem uma associação com a classe *Vehicle*, que determina qual veículo irá perfazer aquela rota específica. Uma rota também possui uma lista de clientes que são atendidos. Esse atendimento é abstraído pela classe *ClientRoute* que indica qual cliente foi atendido por uma determinada rota, qual a posição deste cliente na rota, o tempo em que ele foi

atendido, que é representado pela classe *ServiceTime*, e quais foram as demandas atendidas. A associação entre *ClientRoute* e as demandas é feita por uma lista, lembrando que, cada objeto do tipo *Demand* representa a demanda por um produto. Esse atendimento da demanda é feito pela classe *DemandMet* que descreve se uma demanda por um determinado produto pertencente a um cliente foi atendida na integralidade ou não. A classe *DemandMet* foi onde captou-se o fracionamento da coleta e entrega. Caso não seja permitido fracionamento ou exista a demanda por um único tipo de produto, a lista de demandas na classe *ClientRoute* terá somente um item. O diagrama de classes que representa a concepção de um plano de roteamento pode ser visto na Figura 24.

O diagrama apresentado na Figura 25 corresponde à apresentação de um padrão de projeto chamado *Facade*. Resumidamente, esse padrão de projeto reúne e simplifica o acesso a uma biblioteca de objetos. Dessa forma, a classe *PDPTWBean* concentra todas as classes, métodos e atributos que são necessários para construir um problema de coleta e entrega com janelas de tempo, realizar a chamada ao sistema em C++ e gerar um plano de roteamento no caso de haver uma solução para o problema. As classes *ClientDAO*, *ProductDAO* e *VehicleDAO* são responsáveis pelo gerenciamento de clientes, produtos e veículos junto ao banco de dados, sendo usadas para consultar esses tipos de objetos. Há outras classes responsáveis pela persistência e gerenciamento de outros objetos, mas não convém detalhá-las. A classe *OSRMHandler* encapsula e tem a responsabilidade de gerenciar a comunicação com o servidor OSRM. Este servidor como mencionado na seção anterior é responsável por retornar as matrizes de distância e tempos além de posteriormente retornar as informações para traçar a rota no mapa disponibilizado pelo OSM. O sistema em C++ é carregado como uma biblioteca do sistema, ou seja, uma DLL como explicado anteriormente. O carregamento do sistema, a atribuição de valores aos parâmetros necessários, chamada à ALNS e captura da solução do problema, caso haja, é responsabilidade atribuída à classe *PDPTWAlns* que esconde toda a complexidade dessa tarefa do restante do sistema. Uma observação importante para futuras extensões do *software web* à outras variantes do PRV é que utilizem também o padrão de projeto *Facade*. Recomenda-se essa abordagem, pois ajuda na manutenção do código além de reunir em um único local o arcabouço para a implementação da nova variante adicionada, o que também diminui a dificuldade para a criação da interface gráfica.

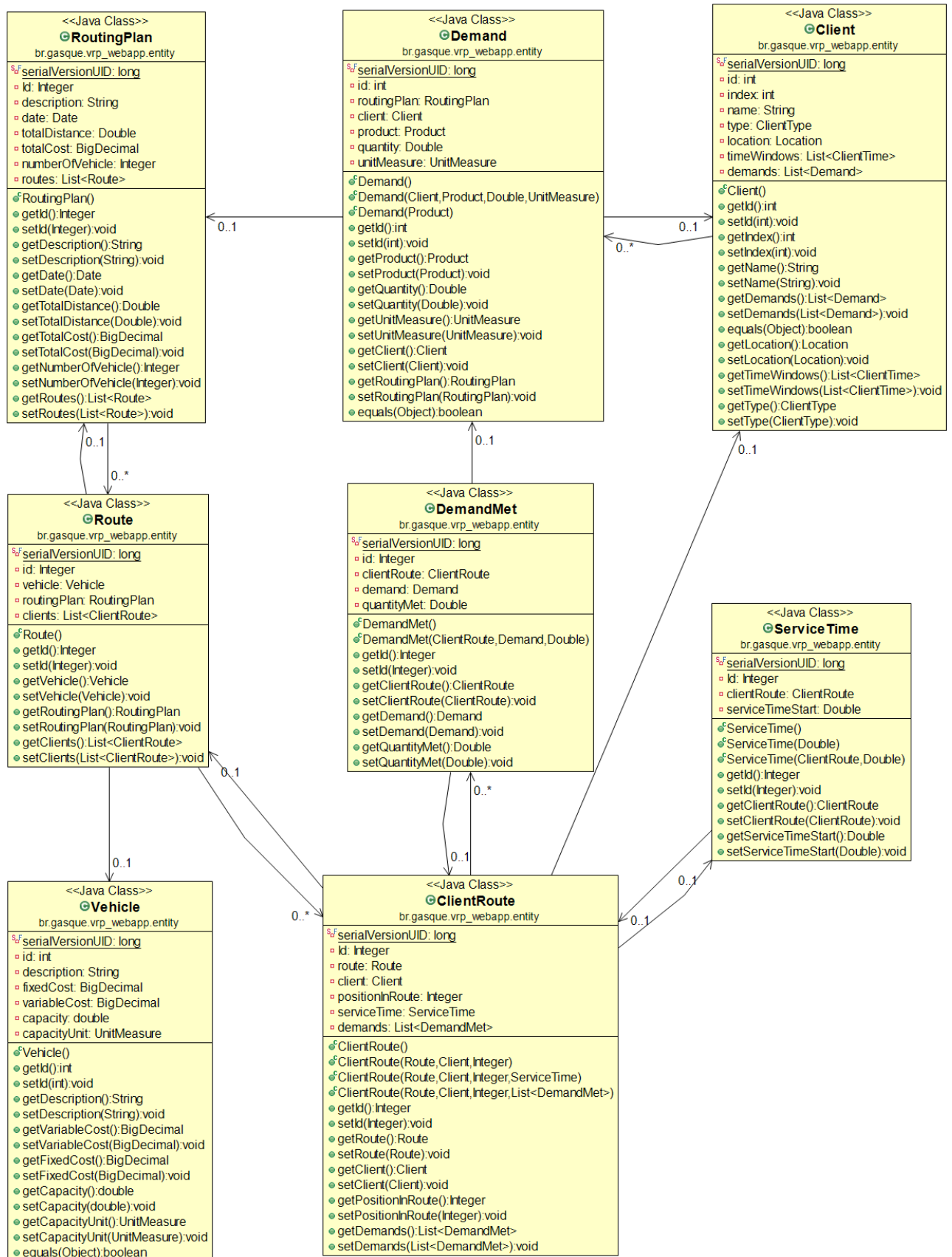


Figura 24: Diagrama de classes do plano de roteamento

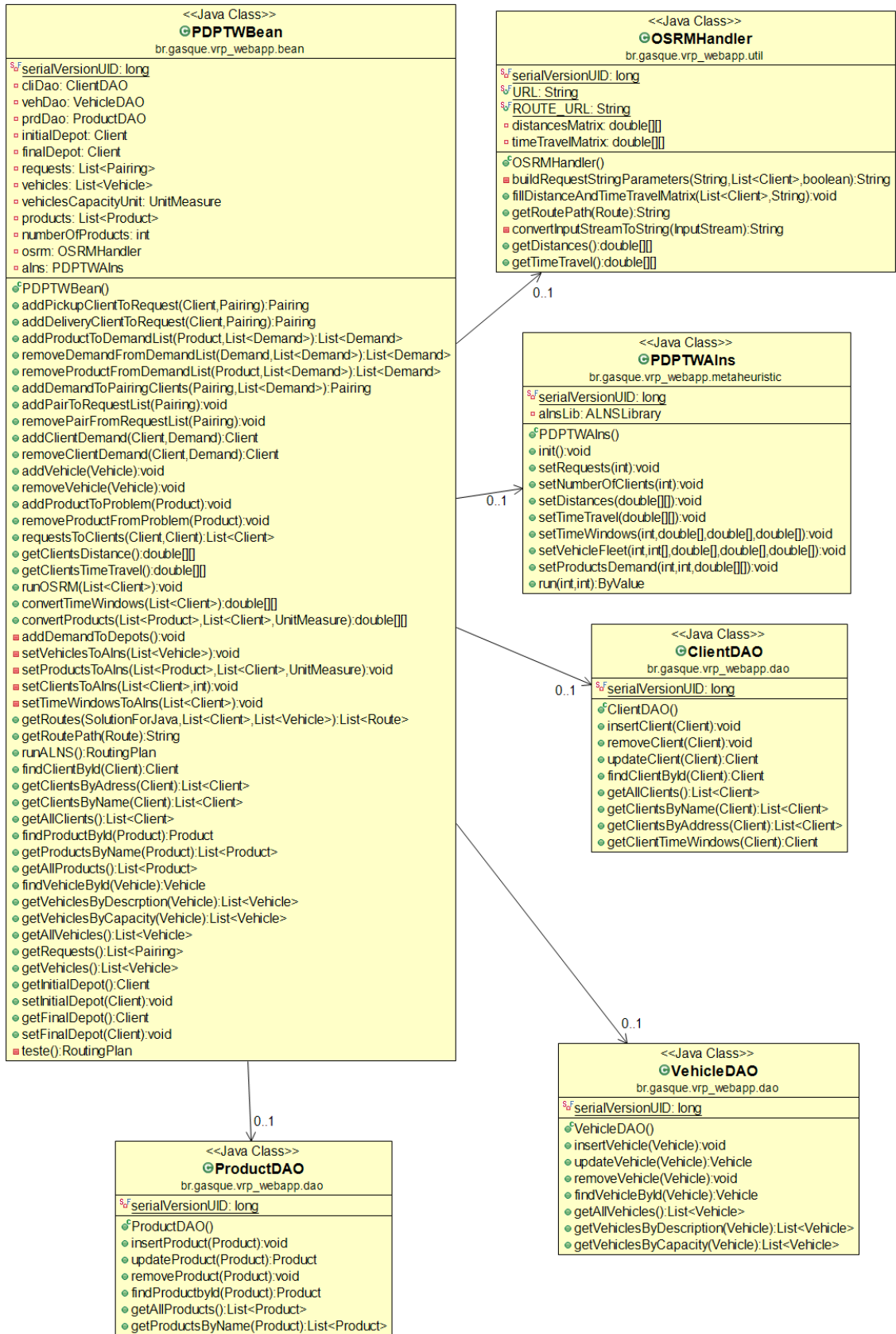


Figura 25: Diagrama de classes.

A Figura 26 apresenta a modelagem dos objetos que tem a responsabilidade de comunicar-se com o sistema C++. Lembrando que essa comunicação é realizada usando uma biblioteca chamada JNA e, no caso de futuras comunicações com outros métodos e sistemas desenvolvidos em C e C++, recomenda-se seu uso. A classe *PDPTWAlns* já foi abordada na descrição do diagrama apresentado na Figura 25. A interface *ALNSLibrary* é a representação dos métodos expostos pela DLL do sistema desenvolvido em C++, onde essa exposição é feita por meio de um arquivo de cabeçalho (*header*). Observe que só é possível acessar um método no sistema desenvolvido em C++ caso o mesmo esteja declarado no arquivo do tipo *header* e também declarado na interface *ALNSLibrary*. Caso haja estruturas de dados (*Struct*) que devam ser acessadas pelo código em Java, estas devem ser declaradas como classes aninhadas *Nested Class*. No caso, duas estruturas de dados foram usadas para obter-se resultados do código em C++, levando a implementação de duas classes *RouteForJava* e *SolutionForJava*, declaradas dentro da interface *ALNSLibrary*.

No diagrama do banco de dados apresentado pela Figura 27 é possível ver o relacionamento das entidades, que são representadas por tabelas. O banco de dados foi modelado de forma a receber novas entidades que venham a representar outras restrições e/ou diferentes atributos relacionados ao PRV. Inicialmente, foram identificados dois tipos de restrições e/ou atributos, que estão vinculadas aos clientes como janelas de tempos e tempos de serviços, e também as que estão ligadas ao plano de rotas como, por exemplo, paridade de visitas. É possível observar que para alguns problemas do PRV é necessária a criação de uma tabela que seja responsável pelo armazenamento da informação no banco de dados. Em outros casos, a restrição pode ser tratada somente pelo método, não sendo necessária sua armazenagem. Pode-se exemplificar a situação mencionada pelas restrições de precedência e paridade que surgem no PCEJT. A primeira pode ser tratada somente pelo método e modelo de resolução. A segunda achou-se por bem armazená-la em uma tabela específica, pois em consultas futuras ao banco de dados para elaboração de relatórios é necessário saber os clientes que fazem parte de uma mesma requisição de coleta e entrega. É imprescindível que em futuras expansões haja a ponderação de quando optar-se por armazenar ou não determinadas restrições e atributos no banco de dados, parte-se da regra que se dados são necessários para o usuário em fases posteriores que não a de obtenção da solução para problema, a persistência dos mesmos deve ser considerada.

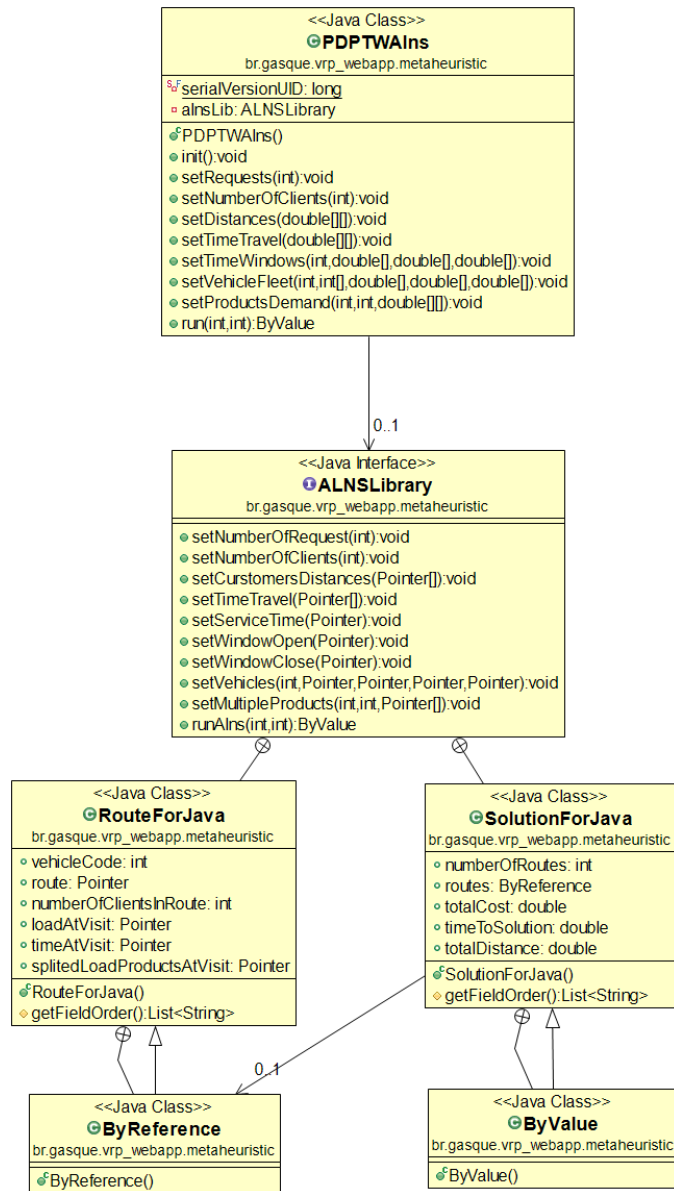


Figura 26: Diagrama de classes.

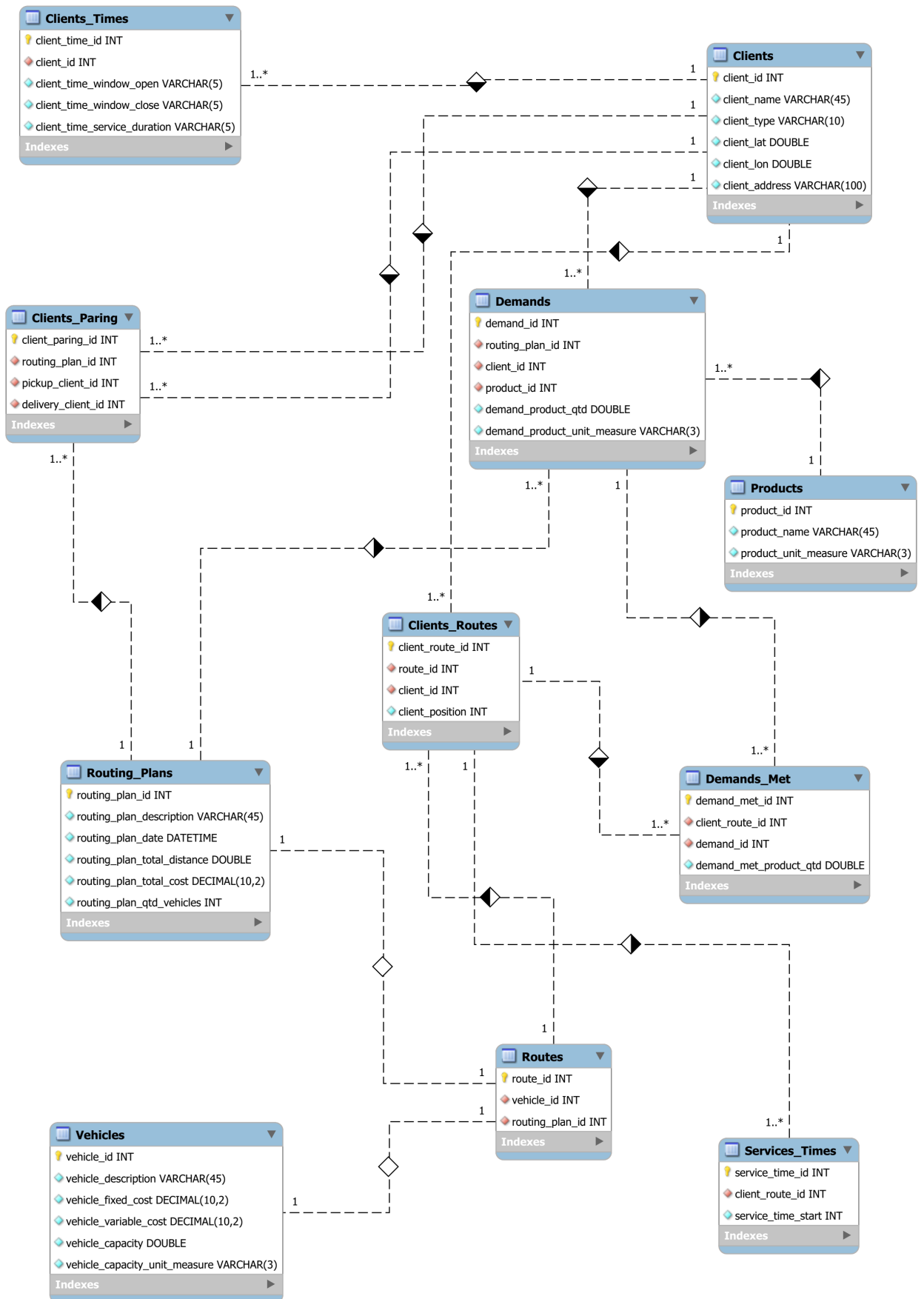


Figura 27: Diagrama de banco de dados EER.

A Tabela 16 apresenta uma descrição breve sobre as tabelas do banco de dados.

Tabela 16: Descrição das tabelas do diagrama do banco de dados.

| Tabela | Descrição |
|-----------------------|--|
| <i>Client</i> | Armazena dados dos clientes cadastrados como endereços entre outros. |
| <i>Client_Times</i> | Guarda informações referentes as janelas de tempos e tempos de serviço dos clientes. |
| <i>Vehicles</i> | Guarda dados sobre os veículos cadastrados como capacidade, custos e descrição. |
| <i>Products</i> | Armazena breve descrição dos produtos e unidade de medida dos mesmos. |
| <i>Routing_Plans</i> | Armazena informações sobre um plano de roteamento, como custo total, quantidade de veículos entre outras. |
| <i>Demands</i> | Armazena a demanda de um produto por um cliente, que pertence a um plano de rotas. Dados como quantidade do produto e unidade de medida são guardados. |
| <i>Routes</i> | Guarda as rotas de um plano de roteamento e qual veículo faz essa rota. |
| <i>Clients_Routes</i> | Armazena que cliente pertence a uma determinada rota e qual sua posição na mesma. |
| <i>Services_Times</i> | Responsável por armazenar o tempo de visita a cada cliente pertencente a uma rota. |
| <i>Demands_Met</i> | Guarda para cada demanda a quantidade atendida por qual rota especificamente. |

6.2.2 Interfaces gráficas

As interfaces gráficas são parte importante de um *software*, pois são elas que detêm a responsabilidade de tornar intuitivo e simples o uso de sistemas complexos, como por exemplo, métodos de solução e modelos matemáticos para problemas de roteamento de veículos. As interfaces do *software web* estão divididas em cadastros simples e gerenciamento do problema, no caso o PCEJT. As primeiras são responsáveis por gravar, remover, alterar e consultar clientes, produtos e veículos no banco de dados. A segunda é responsável por montar uma instância do PCEJT, ou seja, definir quem são os clientes a serem atendidos, quais as demandas por quais produtos e composição da frota de veículos, além de visualizar a solução e também realizar todas as ações desempenhadas pelos cadastros básicos. A seguir, descrevem-se as interfaces gráficas do sistema desenvolvido.

A Figura 28 apresenta o cadastro de um cliente, onde pode-se selecionar sua posição diretamente no mapa e buscar posteriormente seu endereço e posições geográficas, ou fornecer as posições geográficas como latitude e longitude e buscar seu endereço. É possível dizer se o cliente possui ou não janelas de tempo e também assinalar se o mesmo é um depósito ou um nó. Por meio do botão Finalizar, gravam-se os dados no banco de dados. O botão Limpar apaga todos os campos preenchidos até o momento. A Figura 29 contém os campos e a listagem para realização de consulta aos clientes cadastrados, havendo a possibilidade de realizar consulta por nome e por código. Também é possível realizar a alteração e remoção de clientes por meio da listagem. As Figuras 28 e 29 correspondem à mesma tela, sua separação deu-se somente por motivos de

clareza para explicação.

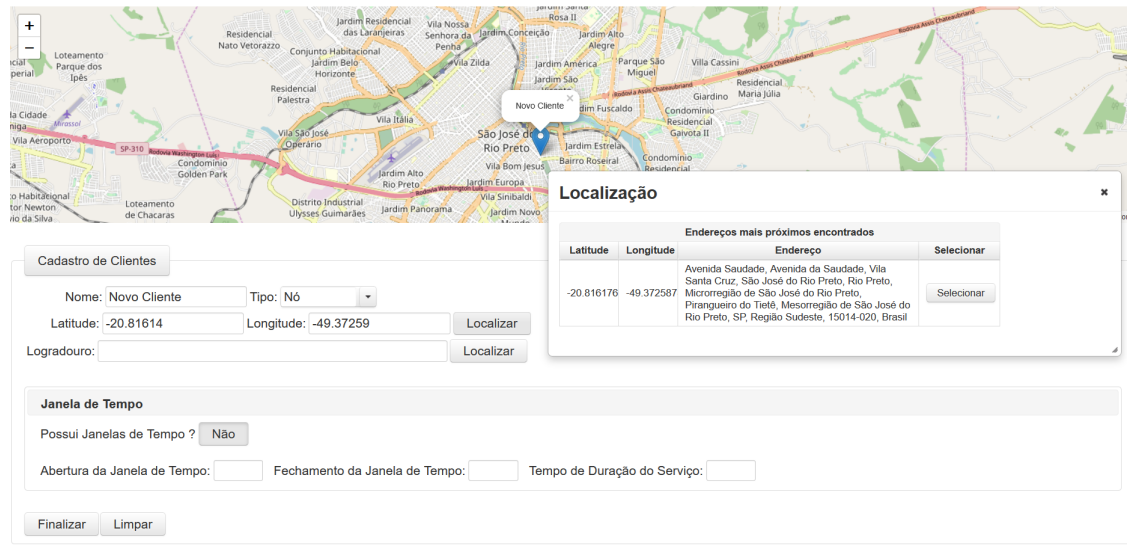


Figura 28: Tela de cadastro de clientes.

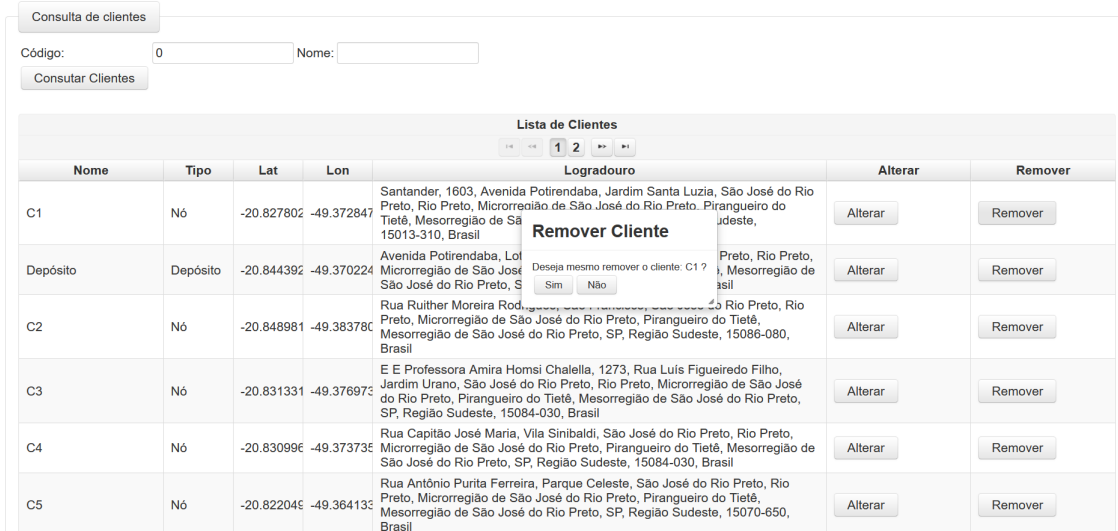


Figura 29: Tela de consulta de clientes.

O cadastro de produtos pode ser visualizado pela Figura 30. Para realizar a gravação de um produto no banco de dados somente é necessário informar o nome e unidade de medida do mesmo e clicar no botão Finalizar. Os botões Finalizar e Limpar têm as mesmas funções exemplificadas anteriormente e são iguais para todas as outras interfaces gráficas. Também é possível realizar a consulta aos produtos anteriormente cadastrados. Esta pode ser feita utilizando-se o nome ou código do produto e, logo em seguida, os resultados encontrados são apresentados em uma listagem que também é usada para alteração e remoção dos resultados.

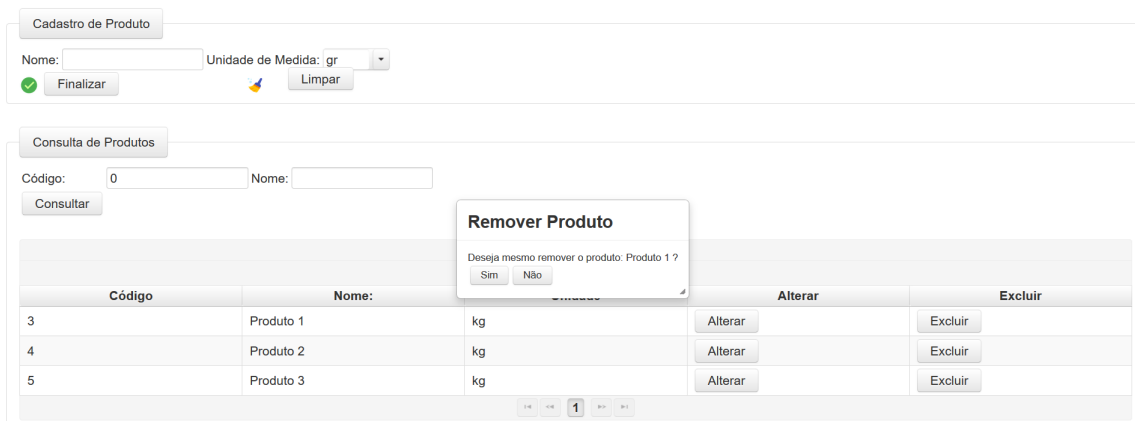


Figura 30: Tela de gerenciamento de produtos.

A interface gráfica apresentada pela Figura 31 é o cadastro de veículos, que é usado para compor a frota de veículo que será usada no PCEJT. Os campos descrição, capacidade, unidade de medida da capacidade, custo fixo e variável devem ser preenchidos para todo veículo a ser cadastrado. Logo abaixo, existe a opção de consultar aqueles que já estão armazenados no banco de dados, utilizando-se do código, descrição ou da capacidade. Após a realização da consulta os resultados são exibidos na listagem, onde há a possibilidade de alteração ou exclusão de qualquer um dos resultados.

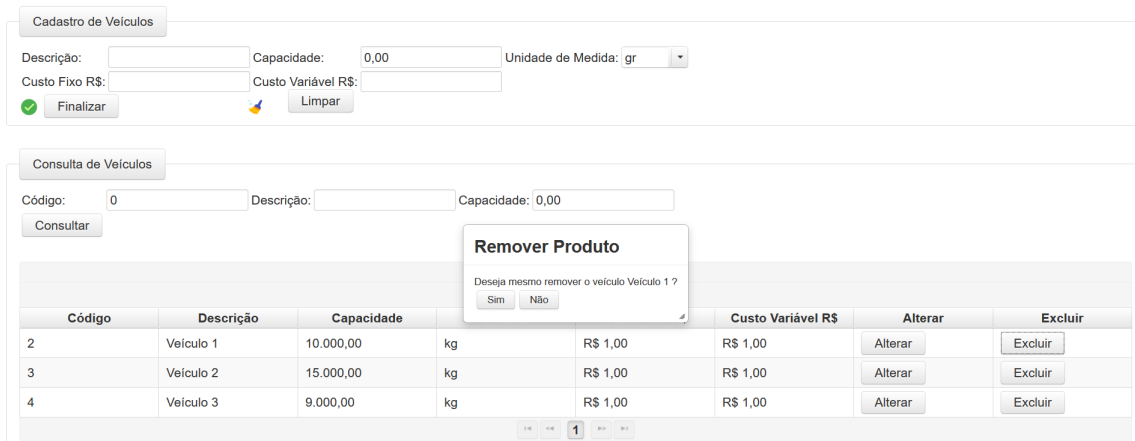


Figura 31: Tela de gerenciamento de veículos.

As Figuras 32 a Figura 39, compõem a tela por onde o usuário é capaz de montar um plano de rotas. Mais precisamente, a aba “Clientes” apresentada pelas Figuras 32 e 33, realiza a montagem das requisições de coleta e entrega. Dessa forma, o usuário é capaz de relacionar um

cliente de coleta a outro de entrega atribuindo-lhes uma demanda, que pode ser composta de um ou múltiplos produtos. Tanto o cliente quanto o produto são vinculados a uma requisição por meio do procedimento denominado “Arraste e solte”, onde os itens são selecionados da lista e arrastados até o campo destinado. Referente à demanda, é possível escolher a quantidade do produto que a compõem bem como a unidade de medida usada no seu transporte. Ao final do procedimento quando o botão “Adicionar” é clicado, a requisição é adicionada ao problema e fica disponível para remoção em uma lista que pode ser vista na aba “Requisições”.

Clientes

Código: 0 Nome: Endereço:

Consultar Clientes

| Lista de Clientes | | | |
|-------------------|--------------|-------------------|---|
| Nome | Latitude | Longitude | Longitude |
| C1 | -20.8278022 | -49.372847497405 | Santander, 1603, Avenida Potirendaba, Jardim Santa Luzia, São José do Rio Preto, Rio Preto, Microrregião de São José do Rio Preto, Pirangueiro do Tietê, Mesorregião de São José do Rio Preto, SP, Região Sudeste, 15013-310, Brasil |
| C10 | -20.84526655 | -49.3660815672521 | Escola Estadual Professor Daud Jorge Simão, Rua Doutor Argemiro Acayaba Toledo, Residencial Cidade Jardim, São José do Rio Preto, Rio Preto, Microrregião de São José do Rio Preto, Pirangueiro do Tietê, Mesorregião de São José do Rio Preto, SP, Região Sudeste, 15080-480, Brasil |

Cliente de Coleta: C1

Cliente de Entrega: C10

Figura 32: Tela de montagem de requisições.

Demanda

Código: 0 Nome:

Consultar

| Lista de Produtos | | |
|-------------------|-----------|---------|
| Código | Nome | Unidade |
| 3 | Produto 1 | kg |
| 4 | Produto 2 | kg |
| 5 | Produto 3 | kg |

| Demanda dos produtos | | | | | |
|----------------------|-----------|--------------------|---------|--------------------|---------|
| Código | Produto | Unidade do Produto | Demanda | Unidade da Demanda | Remover |
| 4 | Produto 2 | kg | 10.0 | gr | Remover |

Adicionar Limpar

Figura 33: Tela de atribuição de demanda à requisição.

A aba “Veículos” representada na Figura 34 demonstra o formulário pelo qual os veículos são selecionados para compor a frota que servirá de base para a elaboração do plano de roteamento. É possível selecionar todos ou somente parte da frota cadastrada.

Códigos: Descrição: Capacidade:

Lista de Veículos

| Código | Descrição | Capacidade | Unidade | Custo Fixo R\$ | Custo Variável R\$ |
|--------|-----------|------------|---------|----------------|--------------------|
| 2 | Veículo 1 | 10.000,00 | kg | R\$ 1,00 | R\$ 1,00 |
| 3 | Veículo 2 | 15.000,00 | kg | R\$ 1,00 | R\$ 1,00 |
| 4 | Veículo 3 | 9.000,00 | kg | R\$ 1,00 | R\$ 1,00 |

| Código | Descrição | Capacidade | Unidade | Custos | Remover |
|--------|-----------|------------|---------|---------------------------------------|--|
| 2 | Veículo 1 | 10.000,00 | kg | <input type="button" value="Custos"/> | <input type="button" value="Remover"/> |

Figura 34: Tela de montagem da frota de veículos a ser usada.

Após a definição das requisições do problema e da frota de veículos disponível, há a necessidade de resolver o problema e apresentar ao usuário as rotas geradas, no caso do sistema em C++ ter encontrado uma solução factível. A tela completa onde as rotas são vistas bem como a sequência das visitas é apresentada pelas Figuras de 35 a 39. Para ilustrar o funcionamento destas telas foi usado um exemplo com cinco requisições, três produtos e três veículos. A priori, o que se observa é uma lista de rotas geradas que ainda não estão plotadas no mapa. Pela listagem inicial é possível visualizar o veículo usado, a rota no mapa e a sequência de visitas aos clientes do problema. Em relação aos dois últimos casos é necessário clicar no botão “Visualizar” para que a rota seja exibida no mapa e a sequência de visitas seja mostrada na tela. A representação das rotas no mapa são observadas pelas Figuras 36 a 38. Enquanto que a sequência de visita aos clientes do problema para uma rota específica é observada na Figura 39.

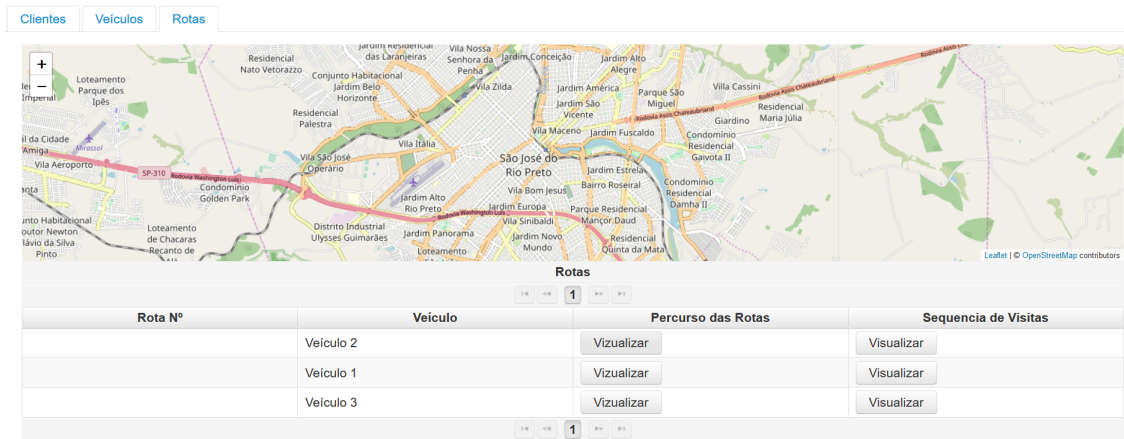


Figura 35: Tela de visualização das rotas geradas.

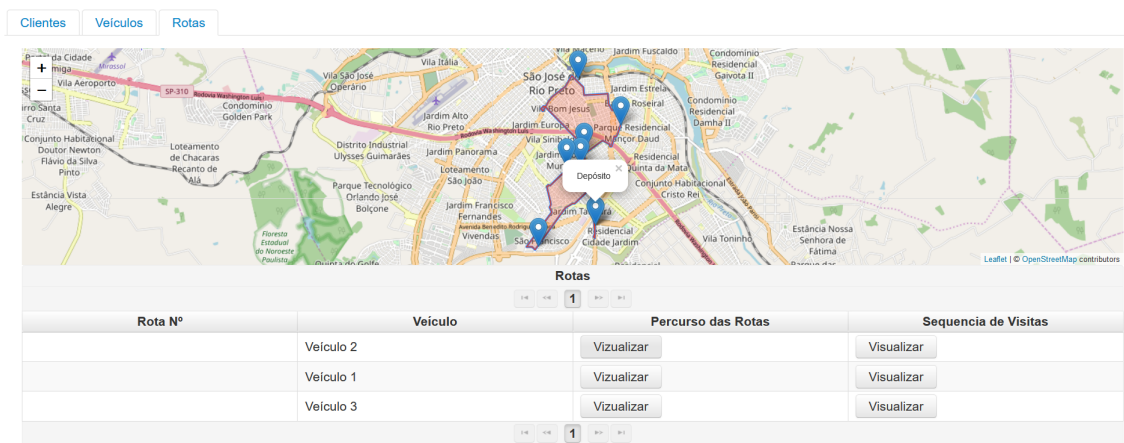


Figura 36: Tela de visualização da rota do veículo 2.

Cientes Veículos Rotas

Rotas

| Rota Nº | Veículo | Percurso das Rotas | Sequencia de Visitas |
|---------|-----------|----------------------------|----------------------------|
| | Veiculo 2 | Vizualizar | Visualizar |
| | Veiculo 1 | Vizualizar | Visualizar |
| | Veiculo 3 | Vizualizar | Visualizar |

Figura 37: Tela de visualização da rota do veículo 1.

Cientes Veículos Rotas

Rotas

| Rota Nº | Veículo | Percurso das Rotas | Sequencia de Visitas |
|---------|-----------|----------------------------|----------------------------|
| | Veiculo 2 | Vizualizar | Visualizar |
| | Veiculo 1 | Vizualizar | Visualizar |
| | Veiculo 3 | Vizualizar | Visualizar |

Figura 38: Tela de visualização da rota do veículo 3.

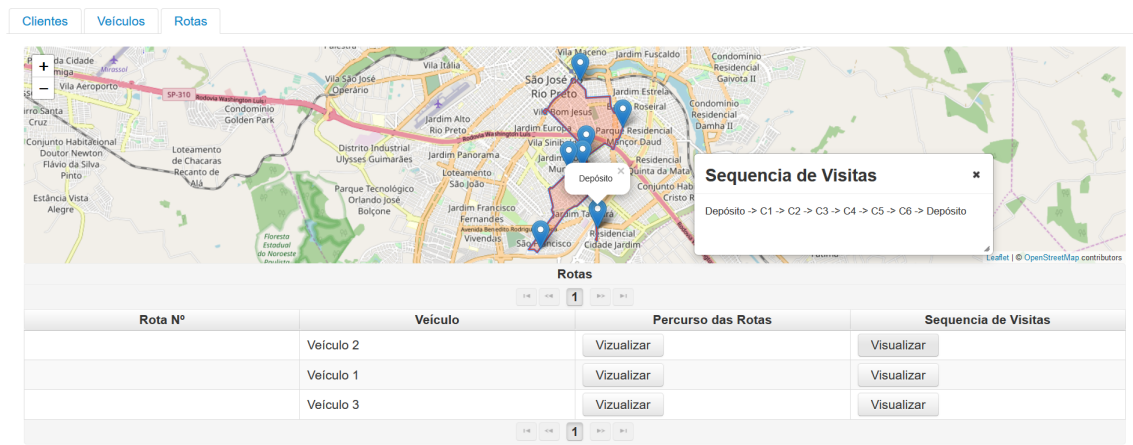


Figura 39: Tela de visualização da sequência de visitas para a rota do veículo 2.

Como primeira versão, observa-se que o sistema desenvolvido é capaz de resolver problemas de roteamento de veículos com coleta e entrega. É perceptível que a ferramenta encontra-se em condições de uso em produção, pois permite a construções de problemas de roteamento de veículos com coleta e entrega de acordo com as especificações do usuário final, além de prover e apresentar visualmente a solução obtida. Por fim, é importar dizer que o sistema pode ser adaptado para uma grande variedade de problemas de roteamento de veículos, tendo como principal ideia a mudança de poucos componentes do *software* além de ser disponibilizado publicamente.

Capítulo 7

Conclusões

Neste trabalho, estudou-se variantes do problema de coleta e entrega com janelas de tempo, que contemplam características práticas com frota heterogênea, múltiplos produtos e coleta e entrega fracionada. Essas variantes têm importantes aplicações no contexto logístico, que vão desde transporte de passageiros até o transporte de mercadorias, seja por via terrestre, marítima ou aérea. Foram desenvolvidos um modelo de dois- e de três-índices, bem como uma meta-heurística ALNS para o PCEJT-1 que contempla frota heterogênea e múltiplos produtos. Posteriormente, o modelo de três-índices e a meta-heurística foram estendidos para o problema PCEJT-2 que abordam o caso com frota heterogênea, múltiplos produtos e coleta e entrega fracionada.

Foram realizados testes computacionais com os modelos e a meta-heurística utilizando-se instâncias da literatura, tanto para o PCEJT-1 quanto para o PCEJT-2. As instâncias por sua vez tiveram que ser adaptadas para incluir frota heterogênea e múltiplos produtos. Os resultados para o PCEJT-1 e PCEJT-2 demonstram que os modelos são capazes de resolver pequenas instâncias de até 15 requisições até a otimalidade, enquanto a meta-heurística proposta consegue obter soluções factíveis de boa qualidade para problemas contendo até 75 requisições, ou seja, 150 clientes, em tempos computacionais aceitáveis. Para as instâncias resolvidas até a otimalidade pelos modelos, para ambas as variantes do PCEJT, a meta-heurística conseguiu equiparar-se em termos do valor das soluções, no entanto, em tempos computacionais menores.

Um *software web* de auxílio à tomada de decisão foi desenvolvido. Esta ferramenta computacional provê aos gestores uma maneira fácil de interagir com métodos científicos que auxiliam à tomada de decisão, como é o caso do método de solução proposto neste trabalho. Além disso, o *software web* serve como interface gráfica a uma variedade de problemas de roteamento de veículos devido a sua estrutura, onde espera-se que novas funcionalidades e restrições usadas para descrever outros tipos de PRVs possam ser facilmente adicionadas. O sistema desenvolvido contemplando o PCEJT será disponibilizado de forma gratuita para a comunidade em geral. O código do *software web* referente a interface gráfica ficará a disposição da comunidade acadêmica de forma livre. Dessa forma espera-se fomentar o uso de métodos e modelos matemáticos que possam ser usados na prática por acadêmicos, indústria e comércio.

A perspectiva para trabalhos futuros envolve o refinamento da ALNS, já que os resultados computacionais apontam nesta direção. Observa-se que as melhorias podem ser realizadas no

método utilizado para fracionamento da coleta e entrega (Haddad et al., 2018) bem como nos mecanismos de remoção e inserção tornando-os mais eficientes (Christiaens e Vanden Berghe, 2018; Vidal et al., 2013a). Existe também, a possibilidade de adaptação dos algoritmos utilizados nesta pesquisa para outros tipos de problemas de roteamento de veículos, principalmente, a problemas denominados ricos, motivados por casos reais. Outra possibilidade para pesquisa futura é a criação de um método híbrido envolvendo um algoritmo do tipo *branch-and-price* e a ALNS desenvolvida. Em relação ao *software web* deve-se voltar esforços para torná-lo mais genérico e intuitivo, além da criação de uma versão para dispositivos móveis.

Referências

- AGUSTINA, D.; LEE, C.; PIPLANI, R. Vehicle scheduling and routing at a cross docking center for food supply chains. *Int. J. Production Economics*, v. 152, p. 29–41, 2014.
- ARCHETTI, C.; SPERANZA, M. G. Vehicle routing problems with split deliveries. *International Transactions in Operational Research*, v. 19, p. 3–22, 2012.
- ARENALES, M. N.; ARMENTANO, V. A.; MORABITO, R.; YANASSE, H. *Pesquisa operacional*. Editora Campus, 2007.
- BALAS, E. Some valid inequalities for the set partitioning problem. Hammer PL, Johnson EL, Korte BH, Nemhauser G, eds. *Studies in Integer Programming, Annals of Discrete Mathematics*, p. 13–47, 1977.
- BALDACCI, R.; BARTOLINI, E.; MINGOZZI, A. An exact algorithm for the pickup and delivery problem with time windows. *Operations research*, v. 59, p. 414–426, 2011.
- BALINSKI, M. L.; QUANDT, R. E. On an integer program for a delivery problem. *Operations Research*, v. 12, n. 2, p. 300–304, 1964.
- BALLOU, R. H. *Business logistics/supply chain management*. 5ª ed. 2004.
- BATTARRA, M.; CORDEAU, J.-F.; IORI, M. Pickup-and-delivery problems for goods transportation. In: TOTH, P.; VIGO, D., eds. *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, p. 161–191, 2014.
- BELFIORE, P.; FÁVERO, L. P. *Pesquisa operacional para cursos de engenharia*. Editora Campus, 2013.
- BENT, R.; HENTENRYCK, P. V. A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers and Operations Research*, v. 33, p. 875–893, 2006.
- BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, v. 22, p. 241–264, 2002.
- BETTINELLI, A.; CESELLI, A.; RIGHINI, G. A branch-and-price algorithm for the multi-depot heterogeneous-fleet pickup and delivery problem with soft time windows. *Mathematical Programming Computation*, v. 6, p. 171–197, 2014.

- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML Guia do Usuário*. 2^a ed. Elsevier, 2006.
- BRAYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, Part I: Route construction and local search algorithms. *Transportation Science*, v. 39, p. 104–118, 2005.
- CHRISTIAENS, J.; VANDEN BERGHE, G. *Slack Induction by String Removals for Vehicle Routing Problems*. Relatório Técnico, 2018.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, v. 12, n. 4, p. 568–581, 1964.
- COELHO, L. C.; LAPORTE, G. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research*, v. 51, n. 23-24, p. 7156–7169, 2013.
- CORDEAU, J.-F.; LAPORTE, G. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, v. 153, n. 1, p. 29–46, 2007.
- CORNILLIER, F.; BOCTOR, F. F.; LAPORTE, G.; RENAUD, J. Discrete Optimization A heuristic for the multi-period petrol station replenishment problem. *European Journal of Operational Research*, v. 191, p. 295–305, 2008.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management science*, v. 6, n. 1, p. 80–91, 1959.
- DESAULNIERS, G.; MADSEN, O. B.; ROPKE, S. The vehicle routing problem with time windows. In: TOTH, P.; VIGO, D., eds. *Vehicle routing: Problems, methods, and applications*, MOS/SIAM Ser Optim, p. 119–159, 2014.
- DREXL, M. Rich vehicle routing in theory and practice. *Logistics Research*, v. 5, n. 1, p. 47–63, 2012.
- FEILLET, D. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR: A Quarterly Journal of Operations Research*, v. 8, p. 407–424, 2010.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of global optimization*, v. 6, n. 2, p. 109–133, 1995.
- FERNEDA, E.; MELLO, B. A.; DINIZ, J. D.; FIGUEIREDO, A. S. An intelligent vehicle routing system for family farming collective organisations. *Int. J. Reasoning-based Intelligent Systems*, v. 5, n. 1, 2012.

- FURTADO, M. G. S.; MUNARI, P.; MORABITO, R. O problema de coleta e entrega na indústria petrolífera: modelagem e método de solução branch-and-cut. In: *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional*, SOBRAPO, 2015, p. 1581–1591.
- FURTADO, M. G. S.; MUNARI, P.; MORABITO, R. Pickup and delivery problem with time windows: A new compact two-index formulation. *Operations Research Letters*, v. 45, n. 4, p. 334–341, 2017.
- GHILAS, V.; DEMIR, E.; VAN WOENSEL, T. An adaptive large neighborhood search heuristic for the Pickup and Delivery Problem with Time Windows and Scheduled Lines. *Computers and Operations Research*, v. 72, p. 12–30, 2016.
- GOLDEN, B. L.; RAGHAVAN, S.; WASIL, E. A. *The vehicle routing problem: latest advances and new challenges*, v. 43. Springer, 2008.
- HADDAD, M. N.; MARTINELLI, R.; VIDAL, T.; OCHI, L. S.; MARTINS, S.; SOUZA, M. J. F.; HARTL, R. Large Neighborhood-Based Metaheuristic and Branch-and-Price for the Pickup and Delivery Problem with Split Loads. *Technical Report - Universidade Federal Fluminense*, p. 1–37, 2018.
- HENKE, T.; SPERANZA, M. G.; WÄSCHER, G. The multi-compartment vehicle routing problem with flexible compartment sizes. *European Journal of Operational Research*, v. 246, p. 730–743, 2015.
- JEPSEN, M.; PETERSEN, B.; SPOORENDONK, S.; PISINGER, D. Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, v. 56, n. 2, p. 497–511, 2008.
- KALLEHAUGE, B.; LARSEN, J.; MADSEN, O. B.; SOLOMON, M. M. Vehicle routing problem with time windows. In: DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M., eds. *Column Generation*, Springer US, p. 67–98, 2005.
- KIRKPATRICK, S.; GELATT JR, C. D.; VECCHI, M. P. Optimization by Simulated Annealing. *Science*, v. 220, n. 4598, p. 671–680, 1983.
- LAHYANI, R.; COELHO, L. C.; KHEMAKHEM, M.; LAPORTE, G. A multi-compartment vehicle routing problem arising in the collection of olive oil in Tunisia. *Omega*, v. 51, p. 1–10, 2015. Disponível em <http://dx.doi.org/10.1016/j.omega.2014.08.007>
- LAPORTE, G.; TOTH, P.; VIGO, D. Vehicle routing: historical perspective and recent contributions. *EURO Journal on Transportation and Logistics*, v. 2, n. 1-2, p. 1–4, 2013.

- LI, H.; LIM, A. A metaheuristic for the pickup and delivery problem with time windows. In: *Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001*, IEEE Comput. Soc, 2001, p. 160–167.
- LU, Q.; DESSOUKY, M. An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science*, v. 38, n. 4, p. 503–514, 2004.
- LU, Q.; DESSOUKY, M. M. A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows. *European Journal of Operational Research*, v. 175, n. 2, p. 672–687, 2006.
- LUXEN, D.; VETTER, C. Real-time routing with openstreetmap data. In: *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '11*, New York, NY, USA: ACM, 2011, p. 513–516 (*GIS '11*,).
Disponível em <http://doi.acm.org/10.1145/2093973.2094062>
- MACGREGOR, J. N.; ORMEROD, T. C. Human performance on the traveling salesman problem. *Perception & psychophysics*, v. 58, n. 4, p. 527–539, 1996.
- MCCAFFREY, J. D. Roulette Wheel Selection Algorithm. <https://jamesmccaffrey.wordpress.com/2017/12/01/roulette-wheel-selection-algorithm/>, Acesso em: 01 de junho 2018, 2017.
- MORABITO, R.; PUREZA, V. Modelagem e simulação. In: *Metodologia de Pesquisa em Engenharia de Produção e Gestão de Operações*, cap. 8, 2012.
- NAGATA, Y.; BRÄYSY, O.; DULLAERT, W. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers and Operations Research*, v. 37, n. 4, p. 724–737, 2010.
- NOWAK, M.; ERGUN, O.; WHITE CHELSEA C., I. I. I. Pickup and Delivery with Split Loads. *Transportation Science*, v. 42, n. 1, p. 32–43, 2008.
- PAREDES-BELMAR, G.; MARIANOV, V.; BRONFMAN, A.; OBREQUE, C.; LÜER-VILLAGRA, A. A milk collection problem with blending. *Transportation Research Part E: Logistics and Transportation Review*, v. 94, p. 26–43, 2016.
- PECIN, D.; CONTARDO, C.; DESAULNIERS, G.; UCHOA, E. New enhancements for the exact solution of the vehicle routing problem with time windows. *INFORMS Journal on Computing*, v. 29, n. 3, p. 489–502, 2017.
- PISINGER, D.; ROPKE, S. A general heuristic for vehicle routing problems. *Computers and Operations Research*, v. 34, n. 8, p. 2403–2435, 2007.

- PISINGER, D.; ROPKE, S. Large Neighborhood Search. In: GENDREAU, M.; POTVIN, J.-Y., eds. *Handbook of Metaheuristics*, Boston, MA: Springer US, p. 399–419, 2010.
- PRESSMAN, R. S. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 2009.
- QU, Y.; BARD, J. F. A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. *Computers and Operations Research*, v. 39, n. 10, p. 2439–2456, 2012.
- QU, Y.; BARD, J. F. The heterogeneous pickup and delivery problem with configurable vehicle capacity. *Transportation Research Part C: Emerging Technologies*, v. 32, p. 1–20, 2013.
- QU, Y.; BARD, J. F. A branch-and-price-and-cut algorithm for heterogeneous pickup and delivery problems with configurable vehicle capacity. *Transportation Science*, v. 49, n. 2, p. 254–270, 2015.
- RAHIMI, M.; DESSOUKY, M. A hierarchical task model for dispatching in computer-assisted demand-responsive paratransit operation. *ITS Journal - Intelligent Transportation Systems Journal*, v. 6, n. 3, p. 199–223, 2001.
- RÍOS-MERCADO, R. Z.; LÓPEZ-PÉREZ, J. F.; CASTRILLÓN-ESCOBAR, A. A grasp for a multi-depot multi-commodity pickup and delivery problem with time windows and heterogeneous fleet in the bottled beverage industry. In: PACINO, D.; VOSS, S.; JENSEN, R. M., eds. *Computational Logistics: 4th International Conference, ICCL 2013, Copenhagen, Denmark, September 25-27, 2013. Proceedings*, Springer Berlin Heidelberg, 2013, p. 143–157.
- ROPKE, S.; CORDEAU, J.-F. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, v. 43, n. 3, p. 267–286, 2009.
- ROPKE, S.; CORDEAU, J.-F.; LAPORTE, G. Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, v. 49, n. 4, p. 258–272, 2007.
- 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.
- SHAPIRO, J. F. *Modeling the supply chain*, v. 2. Thomson Brooks/Cole, 2007.
- SHAW, P. A new local search algorithm providing high quality solutions to vehicle routing problems. *APES Group, Dept of Computer Science, University of . . .*, p. 1–12, 1997.

- SIGURD, M.; PISINGER, D.; SIG, M. Scheduling transportation of live animals to avoid the spread of diseases. *Transportation Science*, v. 38, n. 2, p. 197–209, 2004.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265, 1987.
- TOTH, P.; VIGO, D. *Vehicle routing: Problems, methods and applications*. Second ed. MOS-SIAM Series in Optimization, 2014.
- VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers and Operations Research*, v. 40, n. 1, p. 475–489, 2013a.
- VIDAL, T.; CRAINIC, T. G.; GENDREAU, M.; PRINS, C. Heuristics for multi-attribute vehicle routing problems: a survey and synthesis. *European Journal of Operational Research*, v. 231, n. 1, p. 1–21, 2013b.
- XU, H.; CHEN, Z.-L.; RAJAGOPAL, S.; ARUNAPURAM, S. Solving a practical pickup and delivery problem. *Transportation Science*, v. 37, n. 3, p. 347–364, 2003.
- YAMAN, H. Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming*, v. 106, p. 365–390, 2006.