

WILLIAN GONÇALVES APARECIDO

**MODELAGEM E HEURÍSTICAS PARA PROBLEMAS DE ROTEAMENTO DE  
VEÍCULOS COM ATENDIMENTO SUFICIENTEMENTE PRÓXIMO**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA  
MINAS GERAIS - BRASIL  
2018

**Ficha catalográfica preparada pela Biblioteca Central da Universidade  
Federal de Viçosa - Câmpus Viçosa**

T

A639m  
2018

Aparecido, Willian Gonçalves, 1986-  
Modelagem e heurísticas para problemas de roteamento de  
veículos com atendimento suficientemente próximo / Willian  
Gonçalves Aparecido. – Viçosa, MG, 2018.  
xiii, 86f. : il. (algumas color.) ; 29 cm.

Inclui apêndices.

Orientador: André Gustavo dos Santos.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.83-86.

1. Pesquisa operacional. 2. Problema de roteamento de  
veículos. 3. Radiofrequência - Identificação. 4. Meta-heurística.  
5. Otimização combinatória. I. Universidade Federal de Viçosa.  
Departamento de Informática. Programa de Pós-Graduação em  
Ciência da Computação. II. Título.

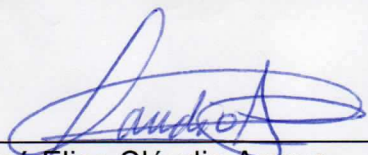
CDD 22. ed. 003.3

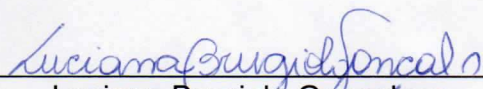
WILLIAN GONÇALVES APARECIDO

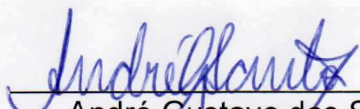
**MODELAGEM E HEURÍSTICAS PARA PROBLEMAS DE  
ROTEAMENTO DE VEÍCULOS COM ATENDIMENTO  
SUFICIENTEMENTE PRÓXIMO**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 28 de setembro de 2018.

  
\_\_\_\_\_  
José Elias Cláudio Arroyo

  
\_\_\_\_\_  
Luciana Brugiolo Gonçalves

  
\_\_\_\_\_  
André Gustavo dos Santos  
(Orientador)

*“Se tudo o que somos e temos recebemos de Deus, do que podemos gloriar-nos?”*  
(São Padre Pio de Pietrelcina)

# Agradecimentos

Agradeço a Deus por mais uma graça alcançada.

A minha esposa Elizabeth, obrigado pelo apoio, paciência e compreensão que, em alguns momentos nem eu me compreendia.

Aos meus pais, Bernadete e Mauro e, aos meus irmãos, Guilherme e Angela, por sempre estarem comigo mesmo estando distantes.

Aos professores do Departamento de Informática e, em especial ao meu orientador André que sem os conhecimentos e experiências compartilhados, este projeto não teria sido concluído.

A Universidade Federal de Viçosa, por me dar a base necessária para realização e execução do projeto.

Aos colegas da Diretoria de Tecnologia da Informação, por me darem suporte no período de estudos.

# Sumário

Lista de Figuras	vi
Lista de Tabelas	ix
Resumo	x
Abstract	xii
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	3
1.2 Contribuições . . . . .	5
1.3 Organização do trabalho . . . . .	5
<b>2 Revisão bibliográfica</b>	<b>7</b>
2.1 Problema do Carteiro Chinês . . . . .	7
2.2 Problema do Carteiro Rural Direcionado . . . . .	9
2.3 Problema de Atendimento de Arcos Suficientemente Próximo . . . . .	11
<b>3 Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo</b>	<b>14</b>
3.1 Definição do problema . . . . .	15
3.1.1 Instância . . . . .	15
3.1.2 Pré-processamento . . . . .	16
3.2 Formulação de programação linear inteira proposta . . . . .	18
3.3 Métodos de solução propostos . . . . .	21
3.3.1 Representação e avaliação da solução . . . . .	21
3.3.2 Heurísticas . . . . .	23
3.4 Resultados e discussão . . . . .	35
3.4.1 Calibração da taxa de destruição da heurística . . . . .	36

3.4.2	Repositório Albaida . . . . .	38
3.4.3	Repositório Madrigueras . . . . .	39
3.4.4	Repositório Random50 . . . . .	41
3.4.5	Repositório Random75 . . . . .	43
3.4.6	Geral . . . . .	44
<b>4</b>	<b>Problema de Roteamento de Veículos com Máximo Atendimento</b>	
	<b>Suficientemente Próximo</b>	<b>47</b>
4.1	Definição do problema . . . . .	48
4.1.1	Instância . . . . .	48
4.2	Formulação de programação linear inteira proposta . . . . .	48
4.3	Métodos de solução propostos . . . . .	51
4.3.1	Função de avaliação . . . . .	51
4.3.2	Heurísticas . . . . .	51
4.4	Resultados e discussão . . . . .	54
4.4.1	Calibração da taxa de destruição da heurística . . . . .	56
4.4.2	Repositório Albaida . . . . .	58
4.4.3	Repositório Madrigueras . . . . .	60
4.4.4	Repositório Random50 . . . . .	60
4.4.5	Repositório Random75 . . . . .	63
4.4.6	Geral . . . . .	64
<b>5</b>	<b>Conclusão</b>	<b>66</b>
5.1	Trabalhos Futuros . . . . .	67
<b>A</b>	<b>Resultados do PRVASP</b>	<b>69</b>
<b>B</b>	<b>Resultados do PRVMASP</b>	<b>76</b>
	<b>Referências Bibliográficas</b>	<b>83</b>

# Lista de Figuras

1.1	Medição por rádio frequência [Ávila et al., 2017]. . . . .	4
2.1	As sete pontes de Königsberg, imagem obtida em Gomes et al. [2009]. . .	8
3.1	Exemplo de instância. Por motivo de clareza, não são mostrados os custos dos arcos. Também não é apresentada a localização dos consumidores, apenas quais os arcos candidatos de cada um. O depósito é o vértice ‘o’, destacado. . . . .	16
3.2	Resultado do pré-processamento de redução de consumidores na instância da Figura 3.1. . . . .	17
3.3	Exemplo de subciclo presente na instância da Figuras 3.1. Para atender o consumidor 1 e continuar a atender os demais, é necessário atravessar por pelo menos duas vezes o arco $(a, b)$ (em destaque). . . . .	18
3.4	Troca de cobertura realizada pelo movimento <i>Reduce</i> . Apenas os arcos candidatos e requeridos são apresentados para facilitar a compreensão do movimento. . . . .	27
3.5	Exemplo do movimento <i>OneDistinct</i> removendo três arcos candidatos sequenciais. Para facilitar o entendimento, apenas os arcos candidatos e requeridos foram apresentados. . . . .	29
3.6	Combinações de rotas disponíveis de acordo com a sequência de atendimento dos consumidores da Figura 3.5. . . . .	29
3.7	Reposicionamento de $h_5$ com o movimento <i>Insertion</i> . Para facilitar o entendimento, apenas os arcos requeridos foram apresentados. . . . .	30
3.8	Movimento <i>exchange</i> entre as rotas superior e inferior, com auxílio do caminho mínimo para reconexão. Para facilitar o entendimento, apenas os arcos candidatos e requeridos foram apresentados. . . . .	32
3.9	Gráfico de médias das distâncias percorridas pelas taxas de destruição utilizadas nos experimentos estatísticos. . . . .	38

3.10	Repositório Albaida: gráfico de média de custos por quantidade de veículos. . . . .	39
3.11	Repositório Albaida: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	40
3.12	Repositório Madrigueras: gráfico de média de custos por quantidade de veículos. . . . .	41
3.13	Repositório Madrigueras: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	41
3.14	Repositório Random50: gráfico de média de custos por quantidade de veículos. . . . .	42
3.15	Repositório Random50: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	43
3.16	Repositório Random75: gráfico de média de custos por quantidade de veículos. . . . .	44
3.17	Repositório Random75: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	44
3.18	Gráfico de média dos custos por repositório. . . . .	45
3.19	Gráfico de tempo médio por repositório. . . . .	46
4.1	Movimento <i>replace</i> com auxílio do caminho mínimo para reconexão . . .	53
4.2	Evolução da qualidade da solução, em relação a última obtida pela execução, com o passar do tempo. Informações de tempo coletadas pelas 10 execuções em todas as instâncias. . . . .	55
4.3	Gráfico de médias dos consumidores atendidos pelas taxas de destruição utilizadas nos experimentos estatísticos. . . . .	58
4.4	Repositório Albaida: gráfico de total de consumidores por número de veículos. . . . .	59
4.5	Repositório Albaida: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	59
4.6	Repositório Madrigueras: gráfico de total de consumidores por número de veículos. . . . .	61
4.7	Repositório Madrigueras: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	61
4.8	Repositório Random50: gráfico de total de consumidores por número de veículos. . . . .	62
4.9	Repositório Random50: gráfico de média dos tempos de execução por quantidade de veículos. . . . .	62

4.10 Repositório <code>Random75</code> : gráfico de total de consumidores por número de veículos. . . . .	64
4.11 Repositório <code>Random75</code> : gráfico de média dos tempos de execução por quantidade de veículos. . . . .	64
4.12 Gráfico de média dos custos por repositório. . . . .	65
4.13 Gráfico de tempo médio por repositório. . . . .	65

# Lista de Tabelas

3.1	Subconjuntos de arcos candidatos pelos quais é possível realizar o atendimento de cada consumidor. . . . .	16
3.2	6 subconjuntos de arcos candidatos dos 6 consumidores após executar o pré-processamento de redução de consumidores . . . . .	17
3.3	Características das instâncias . . . . .	35
3.4	Teste de <i>Scott-Knott</i> , com nível de confiança de 99% . . . . .	38
4.1	Teste de <i>Scott-Knott</i> , com nível de confiança de 99% . . . . .	57
A.1	Resultados das instâncias do repositório Albaida . . . . .	69
A.2	Resultados das instâncias do repositório Madrigueras . . . . .	71
A.3	Resultados das instâncias do repositório Random50 . . . . .	73
A.4	Resultados das instâncias do repositório Random75 . . . . .	74
B.1	Resultados das instâncias do repositório Albaida . . . . .	76
B.2	Resultados das instâncias do repositório Madrigueras . . . . .	78
B.3	Resultados das instâncias do repositório Random50 . . . . .	80
B.4	Resultados das instâncias do repositório Random75 . . . . .	81

# Resumo

APARECIDO, Willian Gonçalves, M.Sc., Universidade Federal de Viçosa, setembro de 2018. **Modelagem e heurísticas para problemas de roteamento de veículos com atendimento suficientemente próximo.** Orientador: André Gustavo dos Santos.

Neste trabalho, foram abordados dois problemas de roteamento para atendimento de arcos suficientemente próximo. O primeiro é o Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP), onde as restrições de atendimento dos consumidores e tamanho das rotas devem ser respeitadas e, em paralelo, objetiva a busca por trajetos de menor custo. Uma aplicação interessante deste problema é a leitura de medidores de água, luz ou gás por rádio frequência através de dispositivos automáticos de leitura, em que veículos equipados com estes dispositivos podem realizar a leitura à distância, e assim, devem passar próximos o suficiente para que os dispositivos colem as informações de cada consumidor. Para resolver este problema, foi proposto um modelo matemático de programação linear inteira e, também, uma meta-heurística híbrida com *Iterated Greedy* e *Variable Neighborhood Descent* (que utiliza quatro estruturas de vizinhanças). O segundo problema analisado é o Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), onde a restrição que força o atendimento de todos os consumidores do PRVASP passa a ser o objetivo, isto é, deve-se atender o maior número de consumidores. Esta é uma variação que condiz com casos práticos, pois nem sempre é possível realizar o atendimento de todos os consumidores, seja por falta de veículos, pela quantidade insuficiente de funcionários responsáveis pela leitura, pela redução de jornada de trabalho, ou ainda, pelo aumento do número de consumidores que devem ser atendidos. Para a resolução do PRVMASP, baseado nos métodos utilizados no PRVASP, foi formulado um modelo de programação linear inteira e foi adicionada à meta-heurística híbrida

proposta uma nova estrutura de vizinhança. Tanto para o PRVASP quanto para o PRVMASP, foram realizados vários testes computacionais com instâncias presentes na literatura, sendo possível comprovar que a meta-heurística desenvolvida IG + VND obteve resultados parecidos com os modelos, validando seu uso em problemas de atendimento de arcos suficientemente próximo e tendo como vantagem o baixo tempo de execução utilizado.

# Abstract

APARECIDO, Willian Gonçalves, M.Sc., Universidade Federal de Viçosa, September, 2018. **Formulation and heuristics for close-enough arc routing problems.** Adviser: André Gustavo dos Santos.

In this work two problems of close-enough arcs covering were studied. The first one is the Close-Enough Cover Vehicle Routing Problem (PRVASP), a routing problem whose objective is to minimize the overall costs of the routes while satisfying two constraints: cover all customers and stay within a route size limit. An interesting application is the reading of water, light or gas meters by radio frequency through automatic reading devices, where vehicles equipped with these devices may read from a distance, and thus, must pass close enough so that the devices can collect information from each consumer. In order to solve this problem, an integer linear programming formulation was proposed and, as well, a hybrid meta-heuristic combining Iterated Greedy and Variable Neighborhood Descent (using four neighborhood structures). In the second problem, the Maximum Cover Close-Enough Vehicle Routing Problem (PRVMASP), the constraint of the PRVASP that forces all consumers to be covered becomes the objective, that is, the largest number of costumers should be covered. This is a variation that is consistent to practical cases, because in some contexts it may not be possible to attend all consumers, either due to lack of vehicles, to the insufficient number of employees responsible for reading, by the reduction of working hours, or even, by the increasing in the number of consumers that must be served. To solve the PRVMASP, based on the methods used in the PRVASP, a linear programming model was formulated and a new neighborhood structure was added to the proposed hybrid meta-heuristic. For both PRVASP and PRVMASP, several tests were carried out with instances present in the literature. It was possible to prove that the developed hybrid metaheuristic IG + VND obtained results similar to the models, validating their use in close-enough arcs covering

problems while taking advantage of the computing time.

# Capítulo 1

## Introdução

Toda a população deveria ter acesso aos serviços essenciais básicos como água tratada, luz e gás [Lei nº 7.783, de 28 de junho de 1989]. No Brasil, os recursos como água, energia elétrica e até mesmo o gás são disponibilizados para a população pelas empresas concessionárias de serviços públicos e as agências reguladoras fiscalizam a prestação destes serviços [Portal Brasil, 2017]. Algumas dessas agências reguladoras são:

- Agência Nacional de Energia Elétrica (ANEEL).
- Agência Nacional de Águas (ANA).
- Agência Nacional do Petróleo (ANP).

A situação do saneamento no Brasil mostra que mais de 35 milhões de brasileiros não têm acesso ao serviço tratamento de água (e mais de 100 milhões ao serviço de coleta esgoto) [Trata Brasil, 2016], sendo necessário a coleta de água feita através da chuva, poços artesianos ou riachos próximos a região, o que acarreta um risco grande à saúde da população. A energia elétrica, embora nem todos tenham acesso através de empresas concessionárias, pode ser adquirida por meios alternativos como painéis solares e geradores a diesel. Já a distribuição do gás liquefeito de petróleo (GLP) é mais comum ser obtida através de microempresas por meio de botijões.

Empresas que adquirem a concessão pública destes serviços utilizam medidores de leitura para contabilizar o consumo, que na maior parte dos casos é lido de porta em porta de forma manual. Normalmente a leitura é realizada mês a mês para todos os consumidores, de modo a garantir uma cobrança correta. Entretanto, esta forma de medição pode apresentar, dentre outros, alguns dos seguintes problemas no momento da coleta:

- Falha humana no instante da leitura e registro da coleta.
- Imóveis desativados por um período e que passam a ser utilizados, onde há ausência de cobrança inicial durante um tempo.
- Ausência de coleta dos dados, por alguns dos motivos:
  - Nenhuma pessoa presente no local para autorizar acesso ao medidor.
  - Área afastada de difícil coleta.
  - Incidentes com animais presentes no local.

Nesses casos, o cálculo é feito por média de meses anteriores, e quando uma leitura real for realizada, pode acontecer uma divergência grande entre as cobranças. No Brasil, desde o ano de 2015, passou a ser particularmente importante a garantia de leituras corretas nas contas de luz, devido às bandeiras tarifárias que acumulam um acréscimo no valor da conta de acordo com o custo da geração de energia [Aneel, 2015]. Para o consumidor, será uma incógnita o valor final, pois dependerá da bandeira tarifária vigente no momento que se consiga realizar a leitura.

Em alguns lugares nos Estados Unidos, já são utilizados leitores de medição automática com tecnologia de identificação por rádio frequência que, instalados em veículos, coletam de forma remota os dados enviados por dispositivos medidores [Shuttleworth et al., 2008], de modo que os leituristas (funcionários das companhias responsáveis pela leitura) não precisam visitar todos os consumidores, bastando apenas passar dentro da área de cobertura de todos os medidores. Com o uso desta tecnologia, há tanto uma garantia por parte do consumidor, que sabe que seu consumo será lido (sua conta, portanto, não será gerada por média de consumos anteriores), como uma economia pela empresa concessionária, que poderá realocar grande parte do pessoal responsável pela leitura em outras áreas, pois este procedimento será feito mais rapidamente e de maneira mais confiável. Isto significa que as vantagens passam desde a melhoria do serviço até a redução de custo.

Com um gasto inicial na capacitação de alguns funcionários que serão responsáveis por realizar as leituras, compra e instalação dos equipamentos e, tendo em contrapartida a redução do número de leituristas e da frota de veículos, existe uma economia a médio prazo devido ao uso da tecnologia de rádio frequência para a empresa. Outra economia a ser obtida é em relação ao trajeto de visitação que atenda a todos os consumidores. A escolha de um menor trajeto que atenda a todos os consumidores, os quais estão dentro do raio de cobertura dos dispositivos por rádio

frequência, economiza tempo e combustível, pois a tecnologia permite que cada consumidor seja atendido por qualquer rua presente em um determinado subconjunto de ruas (subconjunto de ruas que formam a sua área de cobertura) e que o faça de forma mais ágil e eficiente.

O Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP), abordado neste trabalho, tem como exemplo o trajeto realizado por uma quantidade específica de leituristas. Respeitando o limite de distância de cada trajeto, os leituristas saem e retornam com os veículos da garagem da empresa concessionária, de modo a percorrer a menor distância agregada necessária para atendimento de todos os consumidores à distância (tarefa realizada graças à tecnologia de rádio frequência).

Também presente neste trabalho, o Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), uma variação do PRVASP, aborda a situação em que, por motivos técnicos, não é possível garantir a coleta de informações de todos os consumidores (falta de veículos, por exemplo). Neste caso, o objetivo é o atendimento da maior quantidade de consumidores possível com os recursos disponíveis. O limite de distância percorrida de cada veículo e a garagem como ponto de partida e chegada continuam sendo requisitos obrigatórios.

## 1.1 Motivação

A principal aplicação deste problema é construir rotas para a medição de leitura de medidores automáticos que possuem identificador por rádio frequência. Embora utilizado por alguns condomínios para a individualização do consumo de água de cada morador [Batista, 2013], rádio frequência em medidores ainda não está difundida no Brasil.

A rádio frequência é utilizada para a coleta de informações do consumo de cada consumidor à distância, como o consumo de água, luz e gás (Veja a Figura 1.1). Um dispositivo de leitura é instalado em um veículo para coletar os dados enviados pelos dispositivos medidores. O leiturista não precisa visitar cada consumidor para coletar as informações, mas precisa passar dentro do raio de envio do dispositivo, de modo que ele possa realizar seu serviço com maior segurança, confiabilidade e eficiência. O veículo deve passar próximo o suficiente de cada medidor para que todos os dados possam ser lidos.

O atendimento suficientemente próximo dos consumidores gera uma enorme possibilidade de trajetos a serem realizados pelos leituristas. A escolha de um trajeto

mais curto (ou de menor tempo, ou ainda o de menor risco à segurança) possui um atrativo financeiro muito interessante para as empresas responsáveis por este serviço e também para os consumidores, pois será reduzido o custo operacional na coleta dos dados, barateando o serviço como um todo.

O problema de atendimento suficientemente próximo foi inicialmente introduzido pelo trabalho de Shuttleworth et al. [2008], porém com apenas um veículo. Já com múltiplos veículos (PRVASP), o primeiro trabalho foi Ávila et al. [2017]. Devido ao fato do problema ter sido recentemente introduzido na literatura, é pequena a quantidade de trabalhos que o abordam.

A importância prática do problema, complexidade computacional e a grande oportunidade de explorar métodos heurísticos para resolvê-lo são as principais motivações deste trabalho. Outra interessante motivação é a oportunidade de avançar um pouco na literatura, tratando situações onde não é possível atender a todos os consumidores por motivos técnicos ou de pessoal (PRVMASP), de forma que este novo problema se aproxime de casos do dia a dia.

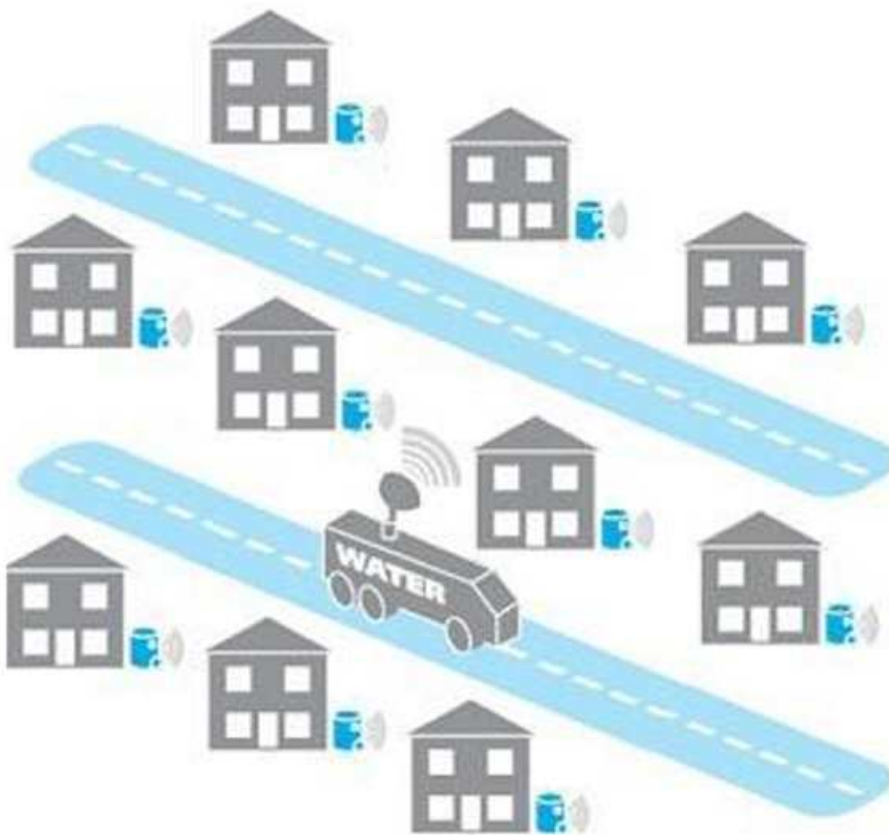


Figura 1.1: Medição por rádio frequência [Ávila et al., 2017].

## 1.2 Contribuições

As principais contribuições deste trabalho são:

- Definição de representação de solução que se adequa melhor aos problemas de roteamento de arcos suficientemente próximos e auxilie, assim, o trabalho feito pelas heurísticas.
- Introdução do Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), um novo problema baseado no Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP).
- Para o PRVASP:
  - Formulação de um modelo de programação linear inteira.
  - Método híbrido utilizando as meta-heurísticas *Iterated Greedy* (IG) e *Variable Neighborhood Descent* (VND).
  - Quatro estruturas de vizinhanças para utilizar em buscas locais, às quais compõem o VND.
- Para o PRVMASP:
  - Formulação de um modelo de programação linear inteira.
  - Método híbrido, resultado de adaptações da heurística proposta para o PRVASP.
  - Nova estrutura de vizinhança para utilizar em buscas locais que se junta às outras quatro para compor o VND.
- Testes e implementações feitas com instâncias contidas na literatura.
- Comparativo entre métodos heurísticos e exatos, comprovando a qualidade dos resultados obtidos do primeiro em relação ao segundo e, mostrando sua superioridade em relação ao tempo de execução no geral.

## 1.3 Organização do trabalho

Este trabalho está dividido em 5 capítulos, incluindo esta introdução.

O Capítulo 2 é composto de uma revisão de trabalhos relacionados aos problemas abordados.

O Capítulo 3 trata do Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP), onde o problema é propriamente definido, um modelo de programação inteira é apresentado e métodos de solução são propostos para a resolução de forma heurística. Os resultados contidos na literatura junto com os resultados obtidos são discutidos no final.

O Capítulo 4 trata do Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), uma variação do PRVASP. Uma definição detalhada do problema é apresentada. Para resolução do problema, um modelo de programação inteira é proposto assim como métodos heurísticos de solução. Os resultados obtidos são apresentados e são feitas análises sobre esses.

Finalmente, encerra-se a dissertação no Capítulo 5, com as principais conclusões do trabalho e as direções de pesquisas futuras.

# Capítulo 2

## Revisão bibliográfica

Neste capítulo é realizada uma revisão dos problemas relacionados à área de roteamento de arcos, de modo a possibilitar um entendimento mais adequado e, assim, contextualizar a área dos problemas de atendimento à distância abordados nesta dissertação. Dentre os problemas de atendimento de arcos destacam-se o Problema do Carteiro Chinês, o Problema do Carteiro Rural Direcionado e o Problema de Atendimento de Arcos Suficientemente Próximo.

### 2.1 Problema do Carteiro Chinês

No Problema de Roteamento de Arcos (PRA), o objetivo é obter o menor custo do trajeto capaz de percorrer um subconjunto de arcos específicos, com ou sem restrições.

O primeiro trabalho sobre PRA que se tem referência é sobre as sete pontes de Königsberg<sup>1</sup> (ilustrado na Figura 2.1), onde buscava-se responder se existia um caminho fechado capaz de atravessar todas as pontes sem repeti-las. Leonhard Euler, um matemático suíço provou em 1736 que não havia possibilidades que satisfizesse aquele caso em particular [Euler, 1736] e, 137 anos mais tarde, Hierholzer & Wiener (em 1873) reforçaram a inexistência do circuito euleriano<sup>2</sup> através das propriedades de conectividade e unicursalidade [Gomes et al., 2009].

Tendo que atravessar pelo menos uma vez cada rua da cidade em busca do percurso de menor custo, o Problema do Carteiro Chinês (PCC) é um problema enfrentado por carteiros que necessitam partir dos correios com as correspondências, entregá-las e retornar aos correios. O problema apareceu pela primeira vez em artigo

---

<sup>1</sup>Desde 1946 denominada Kaliningrado, Rússia

<sup>2</sup>Percurso que se visita cada aresta uma única vez, retornando à origem no final.



Figura 2.1: As sete pontes de Königsberg, imagem obtida em Gomes et al. [2009].

de Mei-Ko publicado em uma revista matemática chinesa [Mei-Ko, 1962]. Segundo Eiselt et al. [1995a], Meigu Guan (ou Kwan Mei-Ko) passou um tempo trabalhando nos correios durante a Revolução Cultural Chinesa.

Mais tarde, no trabalho Edmonds [1965], Jack Edmonds, que estava no grupo de Pesquisa Operacional do Alan J. Goldman no Escritório Nacional de Padrões dos EUA (agora NIST), aceitou a sugestão de Goldman e utilizou o nome Problema do Carteiro Chinês pela primeira vez por saber do trabalho de Kwan [Pieterse & Paul, 2014].

Diversos problemas como roteamento da patrulha de polícia, caminhão de lixo para coleta de resíduos, dentre outros, podem ser modelados sobre o PCC. Há muitas variações do problema na literatura, sendo algumas citadas por Mohammaditabar [2012]:

**Não direcionado** Tratando cada rua como sendo de mão-dupla, o PCC em grafo não direcionado pode ser resolvido de forma eficiente em tempo polinomial. [Mei-Ko, 1962].

**Direcionado** Caso o grafo direcionado seja fortemente conectado, com cada uma de suas arestas possuindo direção, o grafo possui uma rota para o carteiro. Encontrar um trajeto que atenda todos os arcos do grafo direcionado com percurso mínimo pode ser resolvido em tempo polinomial [Edmonds & Johnson, 1973]. Também podemos classificar como Problema do Carteiro Chinês Direcionado um caso especial do Problema do Caixeiro Ventoso ou Íngrime<sup>3</sup>,

<sup>3</sup>Ventoso ou Íngrime pois às vezes está subindo a rua e outras descendo, ou ainda, às vezes está andando a favor do vento, outras contra ele [Guan, 1984]

em que o custo das arestas dependem da direção de travessia (podendo o custo ser ou não inversamente proporcional) e sendo modelado em um grafo direcionado (como visto em Martínez [2008]).

**Misto** É um problema NP-completo onde o grafo possui tanto arcos direcionados como também bi direcionados.

**Capacitado** Uma frota de veículos capacitados deve atender às demandas dos arcos de um grafo, de modo que um veículo possa atender um subconjunto de arcos (que possuem demandas não negativas) sem exceder sua capacidade enquanto minimizam a distância total percorrida. A coleta de lixo pelos caminhões é um exemplo de aplicação para o problema.

**Restrição de tempo** Para esta variação do PCC, o percurso do carteiro deve atentar-se não apenas a quantas vezes passou por cada arco, mas também à sequência em que foram visitados. A ordem de visitação é particularmente importante, pois é necessário que se coincida com as janelas de tempo abertas de cada ponto da cidade e, assim, possa-se realizar a coleta/entrega das mercadorias nos horários permitidos.

**Aberto** Nesta categoria, não é necessário retornar ao depósito; uma aplicação seria mapear as ruas de uma cidade, onde não há a necessidade de retornar à origem quando todas já forem percorridas.

**Hierárquico** Os arcos são agrupados por categoria e uma relação de precedência entre eles é definida. Uma aplicação para este problema é a remoção de neve das ruas, onde se prioriza o serviço em regiões mais movimentadas ou mais críticas, como por exemplo, o entorno de um hospital.

**Rural** Ao invés de realizar o atendimento de todos os arcos, apenas um subconjunto destes é solicitado. Um exemplo simples seria o percurso feito pelo carteiro para entregar algumas encomendas na cidade, não sendo necessário percorrê-la integralmente para realizar tal tarefa.

## 2.2 Problema do Carteiro Rural Direcionado

Dado um grafo  $G = (V, A)$ , onde  $V$  representa o conjunto de vértices e  $A$  o conjunto de arcos e, cada arco possui um custo de percurso associado  $c_{ij} \geq 0$ , o bem conhecido Problema do Carteiro Chinês (PCC), consiste em encontrar o menor

trajeto capaz de atravessar cada arco pelo menos uma vez. O Problema do Carteiro Rural (PCR) é um caso mais geral do PCC onde, ao invés de atravessar todos os arcos, é necessário visitar apenas um subconjunto de arcos  $A_r \subset A$ , os quais são chamados de arcos requeridos. Uma das variações do PCR é o Problema do Carteiro Rural Direcionado (PCRD), onde também deve-se atender o subconjunto de arcos requeridos, porém os arcos do grafo possuem sentido de travessia. O PCRD é um problema NP-difícil [Lenstra & Kan, 1981] desde que o grafo não possa ser reduzido a um grafo fracamente conectado, ou seja, o subconjunto de arcos requeridos devem formar uma quantidade de componentes desconectados.

Rotas para coleta de materiais descartados pela população feita por caminhão de lixo, entregas e coletas de mercadorias pelos correios, inspeção das redes elétricas e telefônicas, dentre outras, são algumas das aplicações práticas que também se encaixam no problema PCRD [Christofides et al., 1986].

Em Christofides et al. [1986], foi proposto um algoritmo *Branch and Bound* para a solução exata do PCRD com o uso de relaxação Lagrangeana<sup>4</sup>. Foram realizados testes em instâncias onde os grafos possuem 80 vértices, 179 arcos e 71 arcos requeridos. Também neste trabalho, foi proposta uma heurística para obter o *Upper Bound* para o algoritmo de *Branch and Bound*. Segundo os autores, o procedimento heurístico foi capaz de encontrar 10 soluções ótimas de 22 instâncias e, em média, ficou 1,42% próximo às soluções ótimas.

Outra aplicação do PCRD ocorre na fabricação de placas de circuito impresso, onde busca-se a melhor sequência de operações de inserção entre as máquinas responsáveis pelas montagens das placas. Ball & Magazine [1988] desenvolveram um algoritmo *Branch and Cut* para a sua resolução de forma ótima.

Segundo Corberán et al. [2014], do mesmo modo que o Problema do Carteiro Rural Misto (PCRM)<sup>5</sup> pode ser convertido no Problema do Caixeiro Viajante Generalizado (PCVG), o PCRD também pode [Blais & Laporte, 2003]. Um arco requerido é trocado por um vértice requerido durante a conversão do PCRD em um problema equivalente de roteamento de vértices, e assim haverá um custo de travessia para cada vértice requerido gerado. Neste grafo gerado, cada arco formado por um par de vértices requeridos possui custo de travessia igual ao custo do caminho mínimo que os une. Após este processo, obtém-se como resultado um grafo para o PCVG que a sua resolução resolve o problema original PCRD. Utilizando

---

<sup>4</sup>Técnica onde problemas difíceis são substituídos por problemas simples que, no decorrer de suas execuções, são ajustados por restrições originárias de resultados obtidos da solução de subproblemas de maneira ótima (exata) ou aproximada (heurística)[Gomes, 2008].

<sup>5</sup>Varição do PCR onde o conjunto de arcos é uma combinação de arcos que podem possuir sentido duplo ou único de travessia.

esta metodologia, Carpaneto et al. [1995] resolveram grafos direcionados gerados aleatoriamente que contavam com 5.000 vértices e 50.000 arcos, com até 3.500 arcos requeridos, por um método exato utilizado para o PCVG [Blais & Laporte, 2003].

Corberán et al. [2014] também cita os trabalhos de Ávila et al. [2015b] e Ávila Valverde [2014], onde o Problema de Roteamento Direcionado Generalizado (PRDG), que tem o PCRD como caso particular, é resolvido por um algoritmo exato *Branch and Cut* que foi construído baseado no estudo do poliedro (famílias de desigualdades provaram ser peculiaridades induzidas deste poliedro). Os autores foram capazes de resolver otimamente 36 instâncias geradas aleatoriamente para o PCRD, que possuíam 1.000 vértices, até 3.200 arcos e 553 conjuntos *R-set*<sup>6</sup>. Além disso, 70 instâncias com as mesmas características de Blais & Laporte [2003] foram geradas e resolvidas otimamente em curto tempo computacional.

Para maior aprofundamento no PCRD, um conjunto de pesquisas relacionada e também algumas variações dele podem ser encontrados nos trabalhos de Eiselt et al. [1995b] e Mourão & Pinto [2017].

## 2.3 Problema de Atendimento de Arcos Suficientemente Próximo

A medição das informações de consumo como água, luz e gás dos consumidores de uma cidade deve ser realizada por meio da tecnologia de rádio frequência. Para executar esta tarefa, é vital que se chegue próximo o suficiente de cada medidor. A vizinhança da qual é possível realizar a coleta das informações é formada por um subconjunto de ruas, as quais estão dentro do raio de abrangência do dispositivo de leitura, bastando que pelo menos um leiturista realize a medição por uma dessas ruas.

No Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP), um conjunto de ruas a ser coberto não é identificado, ao invés disso, uma família de subconjuntos de ruas contidas na cidade é definida. Cada subconjunto contém os arcos (ruas) que estão próximos o suficiente do consumidor para seu atendimento. Encontrar rota(s) de menor custo (ou tempo, ou tamanho) que atenda(m) todos os consumidores, considerando como atendido um consumidor que esteja próximo o suficiente de alguma rota, é o objetivo do PRVASP. Este

---

<sup>6</sup>Conjuntos de vértices que formam os componentes fracamente conexos induzidos pelos vértices e arcos requeridos [Ávila Valverde, 2014]

problema está presente na recente revisão bibliográfica da área de roteamento de arcos [Mourão & Pinto, 2017].

Na tese de doutorado apresentada por Drexel [Drexel, 2007], o PRVASP foi parcialmente introduzido. O problema foi definido em grafos direcionados, porém com apenas um veículo e a presença de um depósito não foi considerada. Neste trabalho, o autor analisou que o PRVASP é uma generalização do Problema do Carteiro Rural Direcionado (PCRD), pois quando cada subconjunto de arcos inclui apenas um arco, o problema se reduz ao PCRD e, portanto, é NP-difícil.

A primeira versão do PRVASP surgiu como Problema do Caixeiro Viajante Suficientemente Próximo, proposto por Shuttleworth et al. [2008]. Em parceria com a empresa de logística *RouteSmart Technologies*, trabalharam visando resolver o problema de coleta de leituras de água, gás ou eletricidade por meio de medidores automáticos de leitura com identificação por rádio frequência. Com uma rota que começasse e terminasse no depósito enquanto passasse dentro de um dado raio de distância de cada consumidor, o objetivo era minimizar a distância percorrida (segundo o autor, o raio efetivo de um dispositivo gira em torno de 150 à 300 metros, porém podendo chegar a 381 metros, o que justificava o atendimento à distância). Quatro heurísticas para a resolução do problema foram propostas e trabalhadas em 18 instâncias reais com média de 900 segmentos de ruas e 9000 consumidores cada. As heurísticas contêm duas fases: na primeira, um subconjunto de ruas é selecionado por meio de uma heurística gulosa ou programação linear inteira e, na segunda fase, é encontrada uma rota por meio de uma heurística que começa e termina no depósito atravessando todas essas ruas. Por as heurísticas serem resolvidas em duas fases, o autor reconhece que a solução encontrada pode ser sub-ótima.

Pouco tempo depois, com o nome de Problema de Roteamento de Arco Suficientemente Próximo, Hà et al. [2012] descreveram o modo como elaboraram um conjunto de instâncias que foi utilizado e apresentaram uma formulação de programação linear inteira mista que funcionava da seguinte maneira: inicialmente resolve-se o problema relaxado sem as restrições de conectividade e, enquanto há restrições que violem a conectividade, elas são adicionadas ao modelo. No trabalho seguinte, Hà et al. [2014] apresentaram resultados computacionais com melhor resultado em instâncias (algumas de grande porte) baseadas em grafos direcionados, não direcionados e mistos, por meio de um algoritmo *Branch-and-Cut*.

Utilizando o nome de Problema Generalizado do Carteiro Rural Direcionado, Ávila et al. [2015a] utilizaram dois tipos de variáveis, um associado com o percurso dos arcos e o outro com os serviços necessários exigidos, para duas novas formulações matemáticas juntamente com novas famílias de desigualdades válidas que

são utilizadas no procedimento de inclusão do *Branch-and-Cut*. Associado a uma dessas formulações, foi realizado um estudo de poliedro de soluções. Os algoritmos apresentados pelos autores obtiveram melhor desempenho que os de Hà et al. [2014].

Por fim, no Problema Generalizado do Carteiro Rural Direcionado com Restrição de Distância, Ávila et al. [2017] mudaram a abordagem de resolução do problema, que até então usava apenas um veículo, para múltiplos veículos e, também, adicionaram restrição de distância máxima percorrida. Para esta versão, segundo os autores, dois dos quatro algoritmos *Branch-and-Cut* propostos tiveram boa performance e foram capazes de resolver instâncias com 5 veículos, 196 vértices, 450 arcos e 150 consumidores.

Também no trabalho [Ávila et al., 2017], é apresentada uma descrição de como foi gerado o conjunto de instâncias. Os conjuntos denominados **Random50** e **Random75** foram criados aleatoriamente em um quadrado de 1000x1000, com total de 50 e 75 vértices. Outros dois conjuntos, **Albaida** e **Madrigueras** (cidades espanholas), foram adaptados de instâncias geradas para Problema do Carteiro Rural Ventoso [Benavent et al., 2007]. Em todas as instâncias construídas, os grafos são fortemente conectados e buscou-se encontrar combinações de quantidade de veículos e tamanho máximo da rota permitido, de maneira que as instâncias não fossem inviáveis e nem triviais e, evitando casos em que a solução ótima não utilizasse todos os veículos. Cada instância contém informações sobre os arcos e respectivos custos, vértices, total de consumidores e arcos que cobrem consumidores. **Random50** e **Random75** possuem 24 instâncias e **Albaida** com **Madrigueras** 48 e, após serem processadas as informações referentes a quantidade de veículos e limite da rota, o total de instâncias saltou para 94 e 172, respectivamente.

Até onde se sabe, o PRVASP foi proposto por Ávila et al. [2017] e não há abordagens heurísticas para a sua resolução.

## Capítulo 3

# Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo

O Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP) é um problema NP-difícil, pois é uma generalização do Problema do Carteiro Rural Direcionado [Drexler, 2007]. Neste problema, os veículos necessitam partir do depósito, passar próximo o suficiente de cada consumidor e retornar ao depósito. O atendimento à distância é possível graças à tecnologia de rádio frequência.

Uma aplicação para o problema é realizar o atendimento à distância de consumidores em uma cidade, por meio de dispositivos automáticos de leitura instalados em veículos. Através destes dispositivos, é possível reduzir o custo do percurso pois não é necessário visitar consumidor a consumidor, mas passar em um subconjunto de ruas nas quais os dados deles possam ser coletados.

Neste capítulo é abordado o PRVASP. Inicialmente, na Seção 3.1, é apresentada a definição formal juntamente com uma instância exemplo e uma técnica de pré-processamento para reduzir a quantidade de consumidores a serem considerados. Na Seção 3.2, é detalhado o modelo de programação inteira proposto. As heurísticas desenvolvidas para o problema aparecem na Seção 3.3, onde são apresentadas também a representação de solução (que mais se adequou ao problema) e a função de avaliação utilizadas. Encerra-se na Seção 3.4, onde são apresentados os resultados e realizada uma discussão sobre eles.

### 3.1 Definição do problema

Seja  $G = (V, A)$  um grafo direcionado fortemente conectado,  $V$  o conjunto de vértices, cujo primeiro é considerado o depósito, e  $A$  o conjunto de arcos, representando as vias da cidade. Cada arco  $(i, j) \in A$  possui um custo associado de percurso  $c_{i,j} \geq 0$ . Neste trabalho o custo  $c_{i,j}$  representa a distância de travessia entre os vértices  $i$  e  $j$  através do arco  $(i, j)$ .

Seja  $\mathbb{W} = \{w_1, \dots, w_W\}$  um conjunto de  $W$  consumidores e  $\mathbb{H} = \{H_1, \dots, H_W\}$  uma família de  $W$  subconjuntos de arcos candidatos, cada  $H_w = \{h_w^i, h_w^{ii}, h_w^{iii}, \dots\}$  possuindo os arcos responsáveis por atender o consumidor  $w \in \mathbb{W}$ . Desta família temos o conjunto de todos os arcos candidatos, dado por  $A_R = H_1 \cup H_2 \cup \dots \cup H_W$ , onde os arcos candidatos são um subconjunto dos arcos ( $A_R \subseteq A$ ). Um arco candidato  $h = (i, j) \in A_R$  pode pertencer a mais de um  $H$ , ou seja, atender a mais de um consumidor.

Sendo  $\mathbb{K} = \{1, \dots, K\}$  o conjunto de  $K$  veículos idênticos, o PRVASP consiste em atender todos os consumidores, isto é, passar por pelo menos um arco candidato de cada consumidor, utilizando para isso até  $K$  rotas diferentes. As rotas começam e terminam no depósito, o trajeto realizado por cada veículo não excede o limite  $T_{max}$  de distância percorrida e, por fim, a distância total agregada percorrida pelos veículos deve ser minimizada.

#### 3.1.1 Instância

As instâncias para o PRVASP são compostas por:

- número  $K$  de veículos
- limite  $T_{max}$  de distância percorrida por veículo
- grafo  $G = (V, A)$ , que representa a rede de ruas
- custo de travessia  $c_{i,j} \geq 0$  associado a cada arco do grafo
- conjunto  $\mathbb{H}$  (conjuntos de arcos que atendem os consumidores)

Na Figura 3.1 apresentamos um exemplo gráfico que considera 8 consumidores de uma instância pequena. Relacionada a esta figura temos a Tabela 3.1, onde para cada consumidor deste exemplo temos a representação dos arcos candidatos responsáveis pelo seu atendimento. Por exemplo, o consumidor 2 pode ser atendido por veículos que passam pelos arcos  $(c, d)$  ou  $(d, e)$  e, também, através do arco  $(d, e)$  é possível realizar o atendimento do consumidor 3.

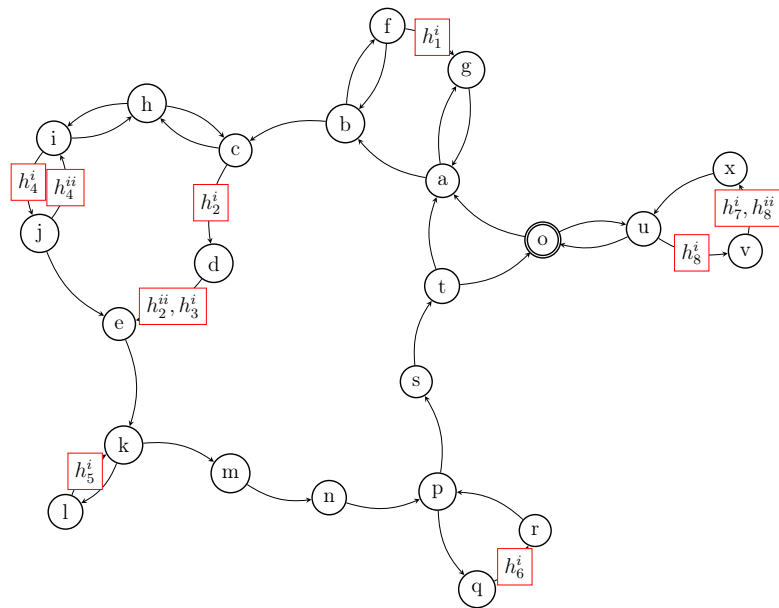


Figura 3.1: Exemplo de instância. Por motivo de clareza, não são mostrados os custos dos arcos. Também não é apresentada a localização dos consumidores, apenas quais os arcos candidatos de cada um. O depósito é o vértice ‘o’, destacado.

Tabela 3.1: Subconjuntos de arcos candidatos pelos quais é possível realizar o atendimento de cada consumidor.

$H_1$	$H_2$	$H_3$	$H_4$	$H_5$	$H_6$	$H_7$	$H_8$
f → g	c → d	d → e	i → j	k → l	q → t	v → x	u → v
	d → e		j → i				v → x

### 3.1.2 Pré-processamento

Primeiramente é descrita a técnica utilizada para a redução do total de consumidores atendidos e, após isso, é descrito como é gerado o caminho mínimo entre todos os pares de vértices do grafo.

#### 3.1.2.1 Redução do total de consumidores atendidos

Para buscar soluções de maneira mais eficiente e conseqüentemente utilizar melhor os recursos computacionais, é realizado um simples pré-processamento nas instâncias logo após sua leitura, de forma a eliminar a redundância direta de atendimento dos consumidores.

A partir da instância representada na Figura 3.1, tomamos como base de análise a família de subconjuntos de arcos responsáveis pelo atendimento dos consumidores demonstrado na Tabela 3.1. Podemos observar que todos os arcos presentes no

subconjunto do consumidor 3 estão contidos no subconjunto do consumidor 2. Isto significa que, caso seja realizado o atendimento do consumidor 3, por consequência o consumidor 2 será atendido. Este fato também ocorre entre os consumidores 7 e 8, mostrando que não é necessário considerar o atendimento dos consumidores 2 e 8, pois uma solução que atenda 3 e 7, também os atende.

O resultado do pré-processamento desta instância pode ser visto na Figura 3.2 e na Tabela 3.2. Em instâncias reais, onde se trabalha com uma grande quantidade de consumidores, a redução dos dados de entrada proporciona um melhor uso do tempo de processamento e também economia de memória, no momento da execução do método de solução proposto (independente de qual seja).

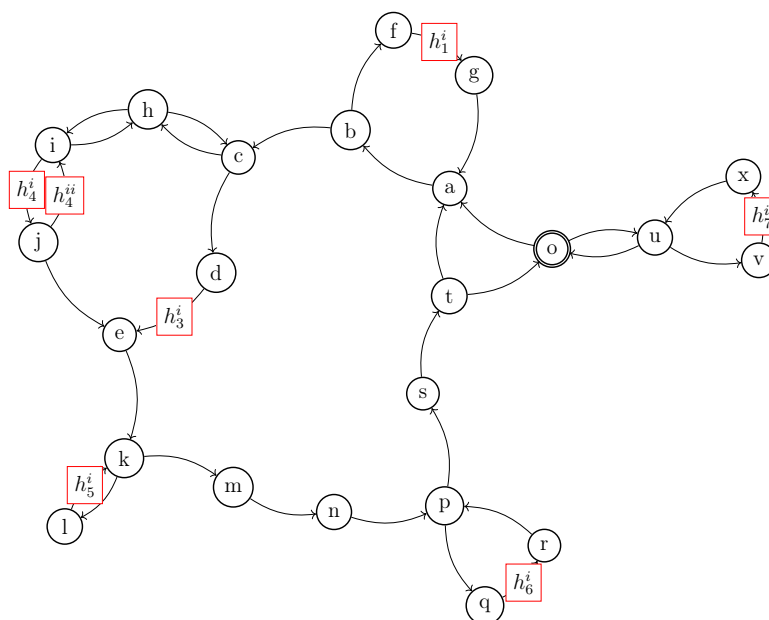
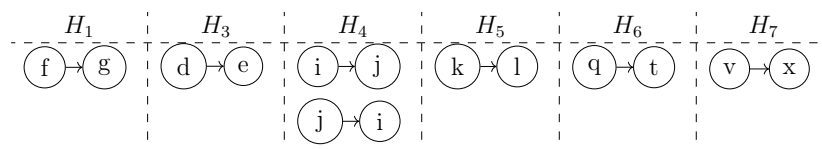


Figura 3.2: Resultado do pré-processamento de redução de consumidores na instância da Figura 3.1.

Tabela 3.2: 6 subconjuntos de arcos candidatos dos 6 consumidores após executar o pré-processamento de redução de consumidores



### 3.1.2.2 Caminho mínimo

As abordagens heurísticas propostas nesta dissertação utilizam diversas vezes o caminho mínimo entre pares de vértices, por isto tais caminhos são previamente

calculados pelo algoritmo de Floyd-Warshall (para detalhes do algoritmo, veja Cormen et al. [2009]).

O algoritmo Floyd-Warshall é usado para construir duas matrizes:

- Caminhos mínimos: para obter a sequência de vértices que formam o caminho mínimo entre cada par deles.
- Distâncias mínimas: armazena a distância mínima juntamente com os consumidores que são atendidos entre cada par de vértices. A informação de quais consumidores são atendidos no trajeto de cada par de vértices é obtida dos arcos candidatos presentes no caminho mínimo.

## 3.2 Formulação de programação linear inteira proposta

Nesta seção é apresentado um modelo completo de programação linear inteira proposto para o PRVASP. A formulação foi baseada na proposta por Hà et al. [2014] com exceção das restrições responsáveis pela eliminação de subciclos, que foi uma adaptação da formulação de fluxo de Gavish & Graves proposta no trabalho Balardino et al. [2016].

O atendimento de alguns consumidores gera a formação de subciclos, como pode ser visto no exemplo da Figura 3.3 e, para este problema, é permitido que haja subciclos, desde que contidos na rota.

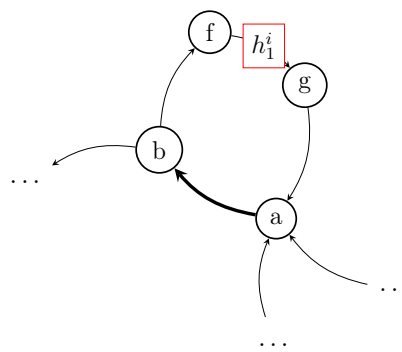


Figura 3.3: Exemplo de subciclo presente na instância da Figuras 3.1. Para atender o consumidor 1 e continuar a atender os demais, é necessário atravessar por pelo menos duas vezes o arco  $(a, b)$  (em destaque).

Para ajudar a realizar o controle do fluxo<sup>1</sup> em cada vértice no modelo, denotamos por  $\delta^-(i)$  os arcos que possuem o vértice  $i$  como destino e por  $\delta^+(i)$  os arcos que o possuem como origem, para todo  $i \in V$ . Formalmente temos  $S \subset V$ , onde os arcos de saída  $\delta^+(S) = \{(i, j) \in A : i \in S, j \in V \setminus S\}$  junto com os arcos de entrada  $\delta^-(S) = \{(i, j) \in A : i \in V \setminus S, j \in S\}$  formam o conjunto de arcos que começam ou terminam em vértices de  $S$ , ou seja,  $\delta(S) = \delta^+(S) \cup \delta^-(S)$ .

A partir dos dados de entrada da instância, são ajustados os valores de algumas constantes presentes no modelo, sendo elas:

- $M$  - limitante superior para restrições de fluxo, definido como  $|V| - 1$ .
- $M'$  - limitante superior para a quantidade de vezes que um veículo pode passar pelo mesmo arco, sendo seu valor definido por  $\min(M, W + 1, |A_R| + 1)$ .

As variáveis utilizadas são descritas abaixo:

- $y_i^k$  - variável binária que vale 1 se o veículo  $k$  passa pelo vértice  $i$ , 0 caso contrário.
- $x_{i,j}^k$  - variável inteira que contabiliza o número de vezes que o veículo  $k$  passa pelo arco  $(i, j)$ .
- $f_{i,j}^k$  - variável auxiliar de controle do fluxo no arco  $(i, j)$  para cada veículo  $k$ .

Para facilitar a notação de controle do fluxo em determinado vértice, as variáveis  $x$  também são apresentadas da seguinte maneira:

- $x^k(\delta^-(i))$  - contabiliza o total de chegadas ao vértice  $i$  pelo veículo  $k$ .
- $x^k(\delta^+(i))$  - contabiliza o total de partidas do vértice  $i$  pelo veículo  $k$ .

A seguir, apresentamos o modelo:

---

<sup>1</sup>Quantidade de vezes que os veículos passam por determinado arco

$$\text{Minimize} \quad \sum_{(i,j) \in A} \sum_{k \in \mathbb{K}} c_{i,j} x_{i,j}^k \quad (3.1)$$

$$\text{s.t.} \quad x^k(\delta^+(0)) \geq 1 \quad \forall k \in \mathbb{K} \quad (3.2)$$

$$x^k(\delta^+(i)) - x^k(\delta^-(i)) = 0 \quad \forall i \in V, \forall k \in \mathbb{K} \quad (3.3)$$

$$\sum_{(i,j) \in H_m} \sum_{k \in \mathbb{K}} x_{i,j}^k \geq 1 \quad \forall H_m \in \mathbb{H} \quad (3.4)$$

$$\sum_{(i,j) \in A} c_{i,j} x_{i,j}^k \leq T_{max} \quad \forall k \in \mathbb{K} \quad (3.5)$$

$$\sum_{j:(i,j) \in A} x_{i,j}^k \geq y_i^k \quad \forall i \in V, \forall k \in \mathbb{K} \quad (3.6)$$

$$\sum_{j:(i,j) \in A} x_{i,j}^k \leq M' y_i^k \quad \forall i \in V, \forall k \in \mathbb{K} \quad (3.7)$$

$$f_{i,j}^k \leq M x_{i,j}^k \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (3.8)$$

$$\sum_{(i,j) \in A} f_{i,j}^k \geq \sum_{(i,j) \in A} f_{j,i}^k + 1 - M(1 - y_i^k) \quad \forall i \in V \setminus \{O\}, \forall k \in \mathbb{K} \quad (3.9)$$

$$x_{i,j}^k \in \mathbb{Z}^+ \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (3.10)$$

$$f_{i,j}^k \in \mathbb{R}^+ \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (3.11)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in V, \forall k \in \mathbb{K} \quad (3.12)$$

A função objetivo do problema é definida pela equação (3.1), a qual busca minimizar a soma das distâncias de todas as rotas percorridas pelos veículos.

A restrição (3.2) garante que o depósito pertença a cada uma das rotas enquanto a restrição (3.3) garante a conservação de fluxos nos vértices, ou seja, a quantidade de vezes que se chega a um vértice é igual à de partidas do mesmo.

O atendimento de cada consumidor fica a cargo da restrição (3.4), que garante o atendimento de algum arco do conjunto de arcos candidatos do consumidor.

A restrição (3.5) é responsável por limitar o tamanho da rota percorrida por cada veículo.

As restrições (3.6) e (3.7) criam uma variável binária que garante que cada veículo  $k$  visite um determinado vértice  $i$  se, e somente se, este veículo atravessar um arco do qual  $i$  faça parte. A restrição (3.8) garante a existência de fluxo apenas em arcos utilizados pelo veículo. A restrição (3.9) permite que um arco seja visitado mais de uma vez, formando subciclos presentes na rota, porém impedindo que eles sejam isolados.

As restrições (3.10), (3.11) e (3.12) definem o domínio das variáveis do pro-

blema.

### 3.3 Métodos de solução propostos

Nesta seção é apresentada a metodologia utilizada para resolver o PRVASP de forma heurística. A representação da solução e a função de avaliação têm influência direta no desempenho das heurísticas. A seguir, são apresentadas as propostas de representação e avaliação e em seguida as heurísticas utilizadas.

#### 3.3.1 Representação e avaliação da solução

Inicialmente é descrita a representação de solução proposta e justificado o seu uso. Em seguida, a função de avaliação adotada também é descrita e justificada.

##### 3.3.1.1 Representação

O espaço de busca das soluções e seu tamanho não é definido unicamente pelo problema abordado, mas também pela representação de solução utilizada e como ela é interpretada. Sendo parte essencial para a resolução do problema, a representação afeta de forma direta o modo como utilizamos as heurísticas e, com isso, tem ligação direta com a qualidade da solução e o desempenho em relação ao tempo de execução [Talbi, 2009].

Nos problemas de roteamento de veículos, uma solução é composta por rotas ( $S = \{R_1, \dots, R_k\}$ ) e cada uma destas rotas normalmente é construída a partir de uma sequência de vértices adjacentes que começam e terminam no vértice de origem.

Utilizando o exemplo da Figura 3.2, supondo uma solução com duas rotas:

$$S = \left\{ R_1 = \{v_o, v_a, v_b, v_f, v_g, v_a, v_b, v_c, v_h, v_i, v_j, v_i, v_h, v_c, v_d, v_e, v_k, v_l, v_k, v_m, v_n, v_p, v_q, v_r, v_p, v_s, v_t, v_o\}, R_2 = \{v_o, v_u, v_v, v_x, v_u, v_o\} \mid v_i \in V \right\}$$

e um movimento *Insertion* (presente no tópico de Busca Local 3.3.2.2) faz com que o vértice  $v_g$  tenha sua posição alterada na rota  $R_1$  para antes do primeiro vértice  $v_b$ ,

assim como mostrado abaixo<sup>2</sup>:

$$(antes) \Rightarrow (depois)$$

$$\{v_o, v_a, \mathbf{v}_b, \mathbf{v}_f, \mathbf{v}_g, \mathbf{v}_a, v_b, v_c, \dots, v_o\} \Rightarrow \{v_o, v_a, \mathbf{v}_g, \mathbf{v}_a, \mathbf{v}_b, \mathbf{v}_f, v_b, v_c, \dots, v_o\}, v_i \in V$$

Este simples movimento fez com que o arco  $(v_f, v_g)$  não fosse mais utilizado, e como efeito colateral fez com que o consumidor 1 deixasse de ser atendido.

Esta representação não é adequada para o problema pois, após qualquer perturbação realizada na rota, deve-se verificar a restrição de atendimento dos consumidores e corrigir a rota, caso necessário.

Estes procedimentos de análise e reparação exigem soluções na maioria das vezes mais complexas, de difícil adaptação em problemas semelhantes e são muito custosos computacionalmente. Por estes motivos, foi utilizada uma representação cuja restrição de atendimento dos consumidores é mais fácil de ser controlada e tais procedimentos não são necessários.

Na representação adotada, as rotas são representadas por sequências de arcos, não de vértices, e utilizamos apenas os arcos que atendem algum consumidor (arcos candidatos). A conexão entre eles é realizada por caminho mínimo (veja Subseção 3.1.2.2), sendo que os arcos no extremo são ligados ao vértice de origem (depósito). Na mesma solução apresentada anteriormente, teríamos a seguinte representação:

$$S = \left\{ R_1 = \{h_1^i, h_4^i, h_4^{ii}, h_3^i, h_5^i, h_6^i\}, R_2 = \{h_7^i\} \mid h_w^\delta \in H_w \right\}$$

### 3.3.1.2 Modificações e consequências

A representação utilizada permite que haja perturbação na ordem de visitação dos arcos requeridos<sup>3</sup> mantendo todos os consumidores atendidos.

Quando ocorre uma mudança na rota, os arcos requeridos da rota são religados através de caminhos mínimos e, quando um destes caminhos religados passa por arcos candidatos, pela nossa abordagem, estes passam a fazer parte da representação da nova solução. No exemplo abaixo, obtido da instância na Figura 3.2, a ordem de atendimento do consumidor 5 da rota  $R_1$  foi alterada, ocasionando um aumento no número de arcos requeridos dentro da rota (os arcos  $h_2^i$  e  $h_2^{ii}$  foram visitados

<sup>2</sup>Foi necessário mover  $v_a$  para manter o correto trajeto através dos arcos e, em destaque, os vértices que tiveram suas posições alteradas

<sup>3</sup>Arcos requeridos são os arcos candidatos presentes na solução

novamente):

$$(antes) \Rightarrow (depois)$$

$$\{h_1^i, h_4^i, h_4^{ii}, h_3^i, \mathbf{h}_5^i, h_6^i\} \Rightarrow \{h_1^i, h_4^i, h_4^{ii}, h_3^i, h_6^i, h_3^i, \mathbf{h}_5^i\}$$

A abordagem de adição dos arcos candidatos, presentes no caminho mínimo da reconexão entre os arcos que sofreram modificação, possibilita uma garantia na contagem de vezes que um consumidor foi atendido, algo que contribui para o aumento de possibilidades de movimentos das heurísticas na busca por novas soluções.

### 3.3.1.3 Função de avaliação

O objetivo do PRVASP é obter a menor distância composta do percurso de todas as rotas, respeitando as restrições de atendimento de todos consumidores e o limite de distância percorrida por cada veículo a um dado valor  $T_{max}$ .

A restrição de atendimento dos consumidores sempre é atendida pois, conforme será visto na descrição das heurísticas de buscas locais, os movimentos feitos pelas heurísticas apenas movem os arcos para outro lugar na rota ou para outra rota, ou ainda trocam arcos por arcos da mesma família de subconjuntos de arcos candidatos. Porém, tais trocas e movimentos não garantem o atendimento da restrição de limite de tamanho da rota. Para buscar atender tal restrição, penaliza-se em 400%<sup>4</sup> o custo da rota que exceda o limite  $T_{max}$ .

A utilização de penalidades foi escolhida pois, além de não requerer complexos procedimentos para manter o tamanho da rota dentro do limite (ou forçar para que tal limite seja respeitado), esta abordagem possibilita navegar em um espaço de soluções viáveis e inviáveis, aproveitando suas melhores características de maneira a escolher a melhor avaliada.

### 3.3.2 Heurísticas

Nesta seção são detalhadas as heurísticas implementadas no presente trabalho, começando pelo algoritmo construtivo, seguido pelas estruturas de vizinhanças das buscas locais que compõem o VND e concluindo com a heurística proposta IG + VND.

---

<sup>4</sup>O valor escolhido possibilita encontrar soluções em todas as instâncias de teste de acordo com as informações contidas na Seção 3.4.

## 3.3.2.1 Heurística construtiva

O Algoritmo 1 mostra um pseudocódigo da heurística construtiva proposta, *BestArc*, explicada a seguir, que retorna uma solução satisfazendo a restrição de atendimento dos consumidores.

**Algoritmo 1:** Pseudocode BestArc

---

**Data:**  $graph = (vertices, arcs), K, T_{max}$   
**Result:** Solution with all customers covered

```

1 routes  $\leftarrow \emptyset$ ;
2 arcsCandidates  $\leftarrow getAllArcsCostumersUncovered()$ ;
3 while arcsCandidates  $\neq \emptyset$  do
4     bestCriterion  $\leftarrow -1$ ;
5     hasValidRoute  $\leftarrow false$ ;
6     for k = 1 to K do
7         foreach arcIn  $\in$  arcsCandidates do
8             foreach position  $\in$  routes[k] do
9                 arcP  $\leftarrow previousArc(position)$ ;
10                arcC  $\leftarrow currentArc(position)$ ;
11                cost  $\leftarrow calcCost(arcP.destiny \rightarrow arcIn \rightarrow arcC.origin)$ ;
12                customers  $\leftarrow totalNewCustomersCovered(arcP.destiny \rightarrow$ 
13                    arcIn  $\rightarrow arcC.origin)$ ;
14                criterion  $\leftarrow customers/cost$ ;
15                costRemoved  $\leftarrow calcCost(arcP.destiny \rightarrow arcC.origin)$ ;
16                isValidRoute  $\leftarrow (route[k].cost - costRemoved + cost) \leq T_{max}$ ;
17                if hasValidRoute and not isValidRoute then
18                    continue;
19                else if criterion > bestCriterion then
20                    bestK  $\leftarrow k$ ;
21                    bestArcIn  $\leftarrow arcIn$ ;
22                    bestPosition  $\leftarrow position$ ;
23                    bestCriterion  $\leftarrow criterion$ ;
24                    hasValidRoute  $\leftarrow isValidRoute$ ;
25                insertInRoute(routes[bestK], bestArcIn, bestPosition);
26                removeUnnecessaryArcs(arcsCandidates, routes);
27 s  $\leftarrow createSolution(routes)$ ;
28 return s;
```

---

**BestArc** A construção da solução por meio deste algoritmo guloso é realizada através de etapas de avaliação e inserção. A cada iteração, o melhor arco candidato avaliado é inserido na solução, repetindo o procedimento até que todos os consumidores tenham sido atendidos.

Inicialmente temos uma lista com todos os arcos candidatos disponíveis (linha 2). Para cada um deles, é determinada a melhor posição de inserção entre cada par

de arcos requeridos consecutivos (contidos na rota) e extremos (depósito) de todas as rotas disponíveis usando como critério a razão entre o número de novos consumidores atendidos e o custo da rota, referentes ao novo trajeto (linhas 11 à 13). Para a escolha do novo arco a ser efetivamente inserido, preza-se primeiramente pela viabilidade da rota (linha 16) e em seguida pelo critério de custo-benefício utilizado (linha 18). Após a escolha, adiciona-se o arco na posição da rota escolhida, juntamente com os arcos candidatos presentes no caminho mínimo das conexões (linha 24), e remove-se da listagem todos os arcos candidatos que não cobrem algum consumidor fora da solução (linha 25), terminando o procedimento quando esta lista fica vazia (linha 3).

### 3.3.2.2 Busca Local

A partir de uma solução inicial, a cada iteração a heurística troca a solução por um vizinho que melhore a função objetivo. A busca termina quando não há candidatos melhores que a solução corrente, significando que um ótimo local foi encontrado.

O Algoritmo 2 mostra um pseudocódigo da busca local e, como pode ser observado, é uma heurística determinística genérica onde os movimentos definem a vizinhança de uma solução.

---

**Algoritmo 2:** Pseudocode local search

---

```
Data:  $graph = (vertices, arcs)$ ,  $s$   
Result: local optima solution  
1 while true do  
    | /* Return best neighbor's solution, the neighborhood varies  
    |   according to the movement like Insertion or others          */  
2    $s_{tmp} \leftarrow doBestMovement(s)$ ;  
3   if  $s.getValue() > s_{tmp}.getValue()$  then  
4   |    $s \leftarrow s_{tmp}$ ;  
5   |   else  
6   |   break;  
7 return  $s$ ;
```

---

Para cada movimento, encontra-se o vizinho de menor valor (obtido por meio da função de avaliação) e então é escolhido o vizinho que dentre todos obteve o melhor resultado. Os movimentos utilizados em buscas locais pela abordagem heurística do PRVSP, explicados a seguir, são classificados como:

- **Movimento intra-rota:** cada vizinho é obtido por movimentos realizados dentro de cada uma das rotas. Os movimentos desta categoria são:

– *Reduce, OneDistinct e Insertion*

- **Movimento inter-rotas:** cada vizinho é obtido por movimentos realizados entre cada par de rotas. O movimento desta categoria é:

– *Exchange*

### Movimentos intra-rota

**Reduce** Este movimento consiste em trocar uma sequência de arcos requeridos<sup>5</sup> por um caminho mínimo dentro de uma rota, sem que algum consumidor fique sem atendimento, com o objetivo de encontrar um trecho que possua uma menor distância de travessia. No Algoritmo 3 é apresentado um pseudocódigo deste movimento.

---

#### Algoritmo 3: Pseudocode movement Reduce

---

```

Data:  $graph = (vertices, arcs)$ ,  $s$ ,  $K$ 
Result: Best solution found
1  $bestDiffValue \leftarrow 0$ ;
2 for  $k = 1$  to  $K$  do                                /* test all the  $K$  routes */
3    $route \leftarrow s.getRoute(k)$ ;                       /* required arcs in vehicle  $k$  */
4   for  $i = 1$  to  $route.size()$  do                   /* test all possible sub-routes */
5     /* With the biggest  $j$ , create  $route_{aux}$  where it's possible to
6       change a sub-path in the route,  $route[i] \rightarrow \dots \rightarrow route[j]$  to
7        $route[i] \rightarrow route[j]$ , that reduce the cost of the new route but
8       preserve the customers covered */
9      $j \leftarrow findBiggestJ(i, route)$ ;
10     $route_{aux} \leftarrow createRoute(route, i, j)$ ;
11     $diffValue_{aux} \leftarrow route.getValue() - route_{aux}.getValue()$ ;
12    if  $diffValue_{aux} > bestDiffValue$  then
13       $bestDiffValue \leftarrow diffValue_{aux}$ ;
14       $bestK \leftarrow k$ ;
15       $bestRoute \leftarrow route_{aux}$ ;
12  $s_{tmp} \leftarrow \emptyset$ ;
13 if  $bestDiffValue > 0$  then
14    $s_{tmp} \leftarrow (s.getRoutes() - s.getRoute(bestK)) \cup bestRoute$ ;
15 return  $s_{tmp}$ ;

```

---

Para que o movimento seja viável e não inviabilize a solução, os consumidores que deixarem de ser atendidos pelo trecho removido devem ser cobertos pelo restante da rota, ou pelo caminho mínimo que vier a ser inserido, ou por demais rotas que não estejam sendo analisadas. De todas as rotas viáveis obtidas pelo movimento, a de menor custo é retornada (ou seja, o vizinho que obteve o melhor resultado).

<sup>5</sup>Arcos candidatos presentes na solução

A Figura 3.4 mostra um exemplo, em que os arcos requeridos  $\{h_9^i, h_{11}^i, h_{12}^i, h_{14}^i\}$  são retirados, e os arcos candidatos  $\{(h_{11}^{ii} = h_{12}^{ii}), h_{11}^{iii}\}$  são conectados na rota. A busca pelo maior trecho (figuras 3.4a à 3.4e) que é substituído pelo caminho mínimo está descrita na linha 5 do Algoritmo 3, enquanto a criação da nova rota (Figura 3.4f) é feita na linha 6. Note que, neste movimento, os consumidores 11 e 12 continuaram a ser atendidos, porém por novos arcos  $(h_{11}^{ii}, h_{11}^{iii})$  e  $h_{12}^{ii}$ , respectivamente e os consumidores 9 e 14 continuaram a ser atendidos por arcos que já se encontravam na rota  $(h_7^i$  e  $h_{14}^{ii})$ .

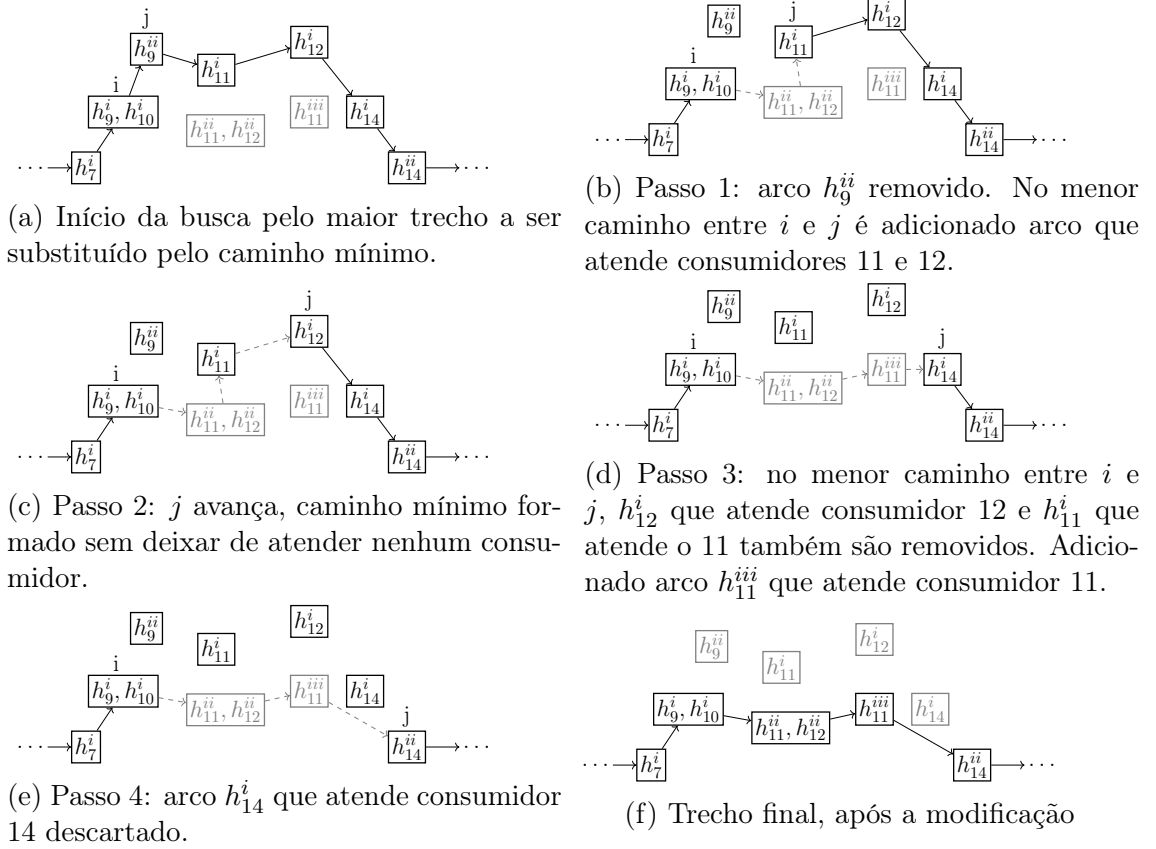


Figura 3.4: Troca de cobertura realizada pelo movimento *Reduce*. Apenas os arcos candidatos e requeridos são apresentados para facilitar a compreensão do movimento.

**OneDistinct** A ideia principal deste movimento é remover um número fixo de arcos sequenciais ao longo de cada rota, identificar os consumidores que ficaram descobertos e atendê-los novamente mediante a melhor conexão possível usando seus arcos candidatos. No Algoritmo 4 é apresentado um pseudocódigo deste movimento. Para identificar a melhor conexão, uma busca exaustiva é realizada por um algoritmo

*backtracking* onde é escolhida a combinação de arcos que obtiver o melhor resultado (linha 6 do algoritmo).

---

**Algoritmo 4:** Pseudocode movement OneDistinct

---

**Data:**  $graph = (vertices, arcs)$ ,  $s$ ,  $K$ ,  $totalRemoved$   
**Result:** Best solution found

```

1  $bestDiffValue \leftarrow 0$ ;
2 for  $k = 1$  to  $K$  do                                /* test all the  $K$  routes */
3    $route \leftarrow s.getRoute(k)$ ;                      /* required arcs in vehicle  $k$  */
4   for  $i = 1$  to  $route.size() - totalRemoved$  do      /* test in all the route */
5      $j \leftarrow i + totalRemoved$ ;
6     /* Remove a slice (from  $i$  to  $j$ ) of the route and reconnect with
7       backtracking all the customers uncovered */
8      $route_{aux} \leftarrow createRouteBacktracking(route, i, j)$ ;
9      $diffValue_{aux} \leftarrow route.getValue() - route_{aux}.getValue()$ ;
10    if  $diffValue_{aux} > bestDiffValue$  then
11       $bestDiffValue \leftarrow diffValue_{aux}$ ;
12       $bestK \leftarrow k$ ;
13       $bestRoute \leftarrow route_{aux}$ ;
12  $s_{tmp} \leftarrow \emptyset$ ;
13 if  $bestDiffValue > 0$  then
14    $s_{tmp} \leftarrow (s.getRoutes() - s.getRoute(bestK)) \cup bestRoute$ ;
15 return  $s_{tmp}$ ;

```

---

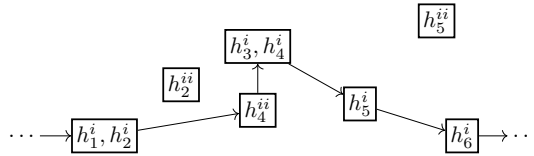
Para aumentar a variabilidade, o trecho substituído na rota resultante deve começar por um arco candidato que não estava presente na rota original, ou seja, o primeiro consumidor que será atendido neste trecho deve ser atendido por outro arco candidato, distinto do que o cobria originalmente (daí o nome da vizinhança). Para gerar as possíveis combinações segundo o critério proposto, o algoritmo *backtracking* realiza as seguintes etapas:

1. identificar os consumidores não atendidos
2. atender os consumidores em todas as ordens possíveis
  - a) Para o primeiro consumidor, testar a cobertura com cada arco candidato do seu subconjunto  $H$ , com exceção do arco que o atendia (caso não haja nenhum outro arco dentro do subconjunto, o processo é interrompido)
  - b) Para os demais consumidores, realizar a cobertura através de cada arco candidato do seu subconjunto  $H$
3. realizar este procedimento até que todos os consumidores sejam atendidos em todas as ordens através de seus subconjuntos de arcos
4. escolher a combinação que obtiver o menor custo de inserção na rota

Um exemplo do movimento pode ser verificado na Figura 3.5 onde, após a remoção de três arcos requeridos seguidos, a melhor combinação dos arcos candidatos que atendem os consumidores 3, 4 e 5 deve ser escolhida. A Figura 3.6 mostra as combinações possíveis obtidas por meio do algoritmo *backtracking* para realizar o atendimento dos consumidores 3, 4 e 5.



(a) Trecho inicial, antes da modificação. Re- (b) Arcos  $h_2^{ii}$ ,  $h_3^i$ (ou  $h_4^i$ ) e  $h_5^{ii}$  removidos. moção de 3 arcos seguidos (de  $i$  à  $j$ ) Consumidores 3, 4 e 5 não são atendidos



(c) Trecho modificado de acordo com a melhor combinação presente na Figura 3.6

Figura 3.5: Exemplo do movimento *OneDistinct* removendo três arcos candidatos sequenciais. Para facilitar o entendimento, apenas os arcos candidatos e requeridos foram apresentados.

$$\begin{aligned}
 (3, 4, 5) &\Rightarrow [\text{não há}] & (4, 5, 3) &\Rightarrow \left[ \begin{array}{l} h_4^{ii} \rightarrow h_5^i \rightarrow \{h_3^i, h_4^i\} \\ h_4^{ii} \rightarrow h_5^{ii} \rightarrow \{h_3^i, h_4^i\} \end{array} \right] \\
 (3, 5, 4) &\Rightarrow [\text{não há}] & (5, 4, 3) &\Rightarrow [h_5^{ii} \rightarrow h_4^{ii} \rightarrow \{h_3^i, h_4^i\}] \\
 (4, 3, 5) &\Rightarrow \left[ \begin{array}{l} h_4^{ii} \rightarrow \{h_3^i, h_4^i\} \rightarrow h_5^i \\ h_4^{ii} \rightarrow \{h_3^i, h_4^i\} \rightarrow h_5^{ii} \end{array} \right] & (5, 3, 4) &\Rightarrow [h_5^{ii} \rightarrow \{h_3^i, h_4^i\} \rightarrow h_4^i]
 \end{aligned}$$

Figura 3.6: Combinações de rotas disponíveis de acordo com a sequência de atendimento dos consumidores da Figura 3.5.

Devido ao tempo gasto com o uso da busca exaustiva, optamos pela remoção de trechos da rota definidos por 4 arcos requeridos. Um detalhe importante é a utilização do caminho mínimo entre todas os arcos candidatos que serão inseridos na rota, não apenas nas extremidades.

**Insertion** O Algoritmo 5 mostra um pseudocódigo do movimento *Insertion*. Neste movimento, um arco requerido é removido e reinserido em outra posição da

rota (linha 8), assim como exemplificado na Figura 3.7. Fazendo este procedimento com todos os arcos da rota e em todas as posições, escolhe-se o vizinho que possua a maior redução na distância percorrida da rota.

---

**Algoritmo 5:** Pseudocode movement Insertion

---

**Data:**  $graph = (vertices, arcs)$ ,  $s$ ,  $K$   
**Result:** Best solution found

```

1  $bestDiffValue \leftarrow 0$ ;
2 for  $k = 1$  to  $K$  do                                     /* test all the  $K$  routes */
3    $route \leftarrow s.getRoute(k)$ ;                             /* required arcs in vehicle  $k$  */
4   for  $i = 1$  to  $route.size()$  do                         /* remove in position  $i$  */
5     for  $j = 1$  to  $route.size()$  do                       /* insert in position before  $j$  */
6       if  $i == (j - 1)$  then
7         continue;
8       /* Create a route that an arc in position  $i$  is removed and
9         insert it before  $j$  */
10       $route_{aux} \leftarrow createRoute(route, i, j)$ ;
11       $diffValue_{aux} \leftarrow route.getValue() - route_{aux}.getValue()$ ;
12      if  $diffValue_{aux} > bestDiffValue$  then
13         $bestDiffValue \leftarrow diffValue_{aux}$ ;
14         $bestK \leftarrow k$ ;
15         $bestRoute \leftarrow route_{aux}$ ;
16  $s_{tmp} \leftarrow \emptyset$ ;
17 if  $bestDiffValue > 0$  then
18    $s_{tmp} \leftarrow (s.getRoutes() - s.getRoute(bestK)) \cup bestRoute$ ;
19 return  $s_{tmp}$ ;

```

---

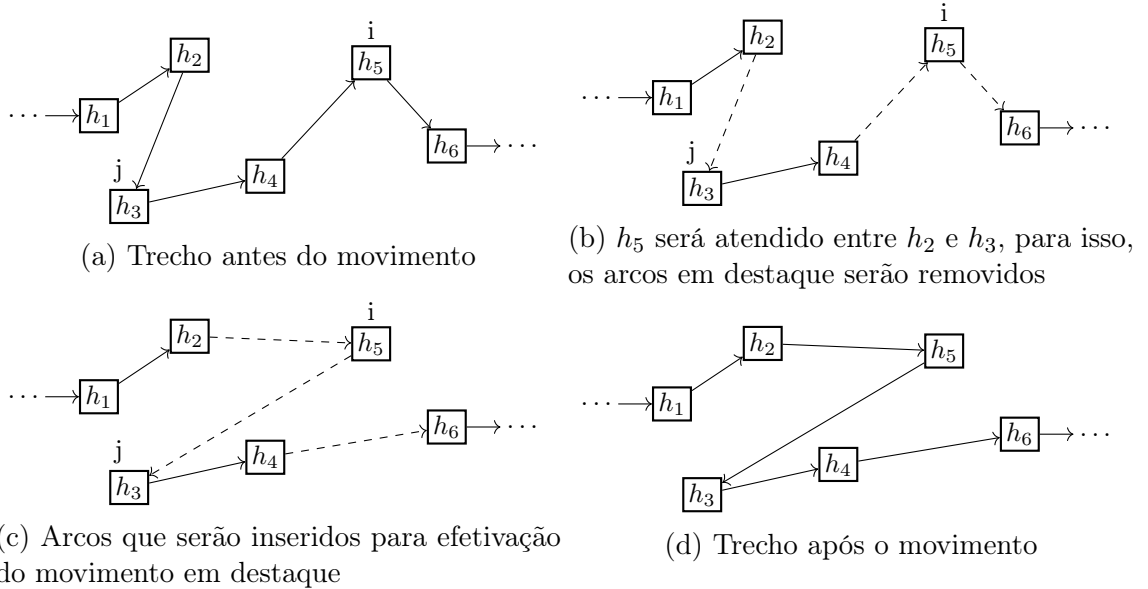


Figura 3.7: Reposicionamento de  $h_5$  com o movimento *Insertion*. Para facilitar o entendimento, apenas os arcos requeridos foram apresentados.

Vale reforçar novamente que, assim como os demais movimentos, este pode

acrescentar outros arcos candidatos à rota, aqueles que fizerem parte do caminho mínimo da reconexão.

### Movimento inter-rotas

**Exchange** Este movimento trabalha entre pares de rotas e é realizado em duas etapas: remoção e correção. Estas duas etapas são realizadas ao longo de cada rota e entre todas as rotas contidas na solução, como pode ser visto no Algoritmo 6.

---

**Algoritmo 6:** Pseudocode movement *Exchange*

---

```

Data:  $graph = (vertices, arcs)$ ,  $s$ ,  $K$ 
Result: Best solution found

1  $s_{tmp} \leftarrow s$ ;
2 for  $kOut = 1$  to  $K$  do                                     /* route to remove an arc */
3    $routeOut \leftarrow s.getRoute(kOut)$ ; /* required arcs in vehicle  $kOut$  */
4   for  $i = 1$  to  $routeOut.size()$  do
5     /* Find the biggest  $j$  for which one can remove
6        $routeOut[i] \rightarrow \dots \rightarrow routeOut[j]$  to  $routeOut[i] \rightarrow routeOut[j]$  in
7        $routeOut_{aux}$ , that uncovering a set of costumers (filled in
8        $customersOut$ ) can be covered by other arc */
9      $j \leftarrow findBiggestJ(i, routeOut, customersOut)$ ;
10     $routeOut_{aux} \leftarrow createRoute(routeOut, i, j)$ ;
11    /* Find only the arc candidates that cover all costumers in
12     $customersOut$  */
13     $arcsCandidates \leftarrow getCandidatesArcs(customersOut)$ ;
14    for  $kIn = 1$  to  $K$  do /* route to insert an arc candidate */
15      if  $kIn == kOut$  then
16        continue;
17       $routeIn \leftarrow s.getRoute(kIn)$ ;
18      for  $m = 1$  to  $routeIn.size()$  do
19        for  $h = 0$  to  $arcsCandidates.size()$  do
20          /* Find the biggest  $n$  that one can remove
21             $routeIn[m] \rightarrow \dots \rightarrow routeIn[n]$  and insert a
22             $arcsCandidates[h]$  and cover all costumers */
23           $n \leftarrow findBiggestN(m, routeIn, arcsCandidates[h])$ ;
24           $routeIn_{aux} \leftarrow$ 
25             $createRoute(routeIn, m, arcsCandidates[h], m)$ ;
26           $s_{aux} \leftarrow (s.getRoutes() - routeOut - routeIn) \cup$ 
27             $routeOut_{aux} \cup routeIn_{aux}$ ;
28          if  $s_{aux}.getValue() < s_{tmp}.getValue()$  then
29             $s_{tmp} \leftarrow s_{aux}$ ;
30 return  $s_{tmp}$ ;

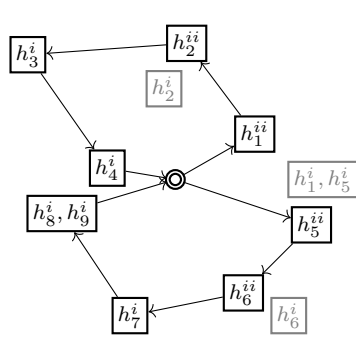
```

---

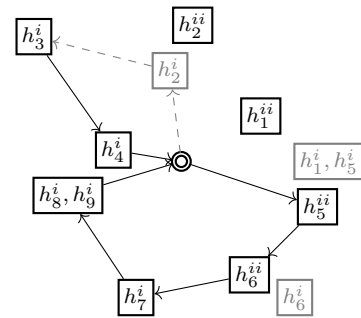
As duas etapas funcionam da seguinte maneira:

- **Etapa de remoção** (linhas 5, 6 e 7): remove-se a sequência com maior quantidade de arcos da primeira rota, de modo que haja pelo menos um arco candidato capaz de cobrir todos os consumidores descobertos, isto é, aqueles que deixaram de ser atendidos com a remoção dos arcos.
- **Etapa de correção** (linhas 14 e 15): testa-se a inserção de cada arco candidato capaz de cobrir consumidores descobertos nas demais rotas, realizando em conjunto o processo de redução de caminho visto no movimento *Reduce* na região onde o arco será adicionado.

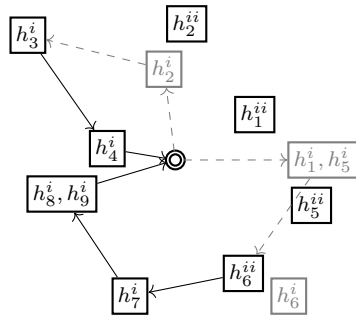
Finalizado o movimento, escolhe-se o vizinho que obteve a melhor combinação de remoção e correção (linha 18).



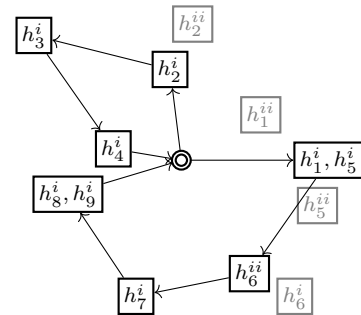
(a) Rotas presentes na solução



(b) Consumidor 1 deixa de ser atendido pela rota superior e deverá ser coberto por outra



(c) Consumidor 1 passa a ser atendido pela rota inferior e houve mudança na cobertura do consumidor 5 pois novo trajeto possui menor distância



(d) Novas rotas, procedimento finalizado

Figura 3.8: Movimento *exchange* entre as rotas superior e inferior, com auxílio do caminho mínimo para reconexão. Para facilitar o entendimento, apenas os arcos candidatos e requeridos foram apresentados.

No exemplo da Figura 3.8, é mostrada uma troca de atendimento feita mediante o movimento *Exchange*. Na etapa de remoção, que é realizada na rota superior (Figura 3.8b), o consumidor 1 deixa de ser atendido pelo arco  $h_1^i$ . Note que isto

causa uma mudança do arco responsável pelo atendimento do consumidor 2 (de  $h_2^{ii}$  para  $h_2^i$ ). Na etapa de correção, que é realizada na rota inferior (Figura 3.8c), o consumidor 1 volta a ser atendido (arco  $h_1^i$ ) e, também ocorre a mudança de atendimento do consumidor 5 (de  $h_5^{ii}$  para  $h_5^i$ ). Um movimento similar ao *Reduce* está presente nas duas etapas, permitindo reduzir a distância percorrida, quando possível, pela reconexão por novos arcos, como é possível verificar o fato ocorrido com os consumidores 2 e 5.

### 3.3.2.3 *Variable Neighborhood Descent*

Segundo Talbi [2009], o algoritmo determinístico *Variable Neighborhood Descent* (VND) usa sucessivas vizinhanças em busca de um ótimo local (Algoritmo 7). Primeiramente, um conjunto de estruturas de vizinhanças (que se complementam) são selecionadas e ordenadas, sendo a estratégia mais popular para a ordenação a que leva em consideração o crescimento da complexidade de cada vizinhança (ou seja, o tamanho das vizinhanças). Quanto mais largas forem as vizinhanças, mais tempo de consumo será utilizado pelo VND. A primeira estrutura é executada enquanto há melhoria na solução, após isso, passa-se para a estrutura seguinte. Se ocorre melhoria, a primeira estrutura torna a ser executada novamente. O algoritmo termina quando nenhuma estrutura de vizinhança consegue melhorar a solução.

---

#### Algoritmo 7: Pseudocode VND

---

**Data:**  $graph = (vertices, arcs)$ ,  $s_0$ ,  
 $localSearches[reduce, oneDistinct, insertion, exchange]$

**Result:** Best solution found

```

1  $s \leftarrow s_0$ ;
2  $l \leftarrow 1$ ;
3 while  $l \leq localSearches.size()$  do
4    $s_{tmp} \leftarrow localSearches[l].bestImprovement(s)$  ;
5   if  $s.getValue() > s_{tmp}.getValue()$  then
6      $s \leftarrow s_{tmp}$ ;
7      $l \leftarrow 1$ ;
8   else
9      $l \leftarrow l + 1$ ;
10 return  $s$ ;
```

---

A abordagem utilizada para o VND foi o *best improvement*, ou seja, cada vez que uma estrutura de vizinhança é explorada, ela é explorada por completo e a melhor solução vizinha é retornada. A sequência dos movimentos utilizada pelo

VND, de acordo com o tamanho das vizinhanças, foi *Reduce*, *OneDistinct*, *Insertion* e *Exchange*.

### 3.3.2.4 *Iterated Greedy*+VND

Segundo Stützle & Ruiz [2018], o algoritmo *Iterated Greedy* (IG) é um método de busca por soluções que executa duas fases principais durante suas iterações: a primeira fase consiste na destruição parcial da solução corrente (retira-se uma quantidade de itens da solução de forma aleatória) e, a segunda fase consiste em reconstruir a solução por meio de um algoritmo construtivo guloso (reparando a solução obtida da fase anterior). Após a execução das fases de destruição e reconstrução, o critério de aceitação decidirá se a nova solução substituirá a atual solução.

IG é um método simples que tem apresentado bons resultados em diversos problemas de otimização combinatória, como *Traveling Salesman Problem* (TSP), *Set Covering Problem* (SCP) e *Permutation Flow Shop Problem* (PFSP), segundo Stützle & Ruiz [2018].

A meta-heurística híbrida do IG + VND apresentada no Algoritmo 8 possui como base o IG e o VND como sendo a terceira fase deste algoritmo. O IG + VND possui IG atuando na diversificação e o VND na intensificação da solução.

---

**Algoritmo 8:** Pseudocode IG + VND

---

```

Data: graph = (vertices, arcs)
Result: Best solution found
1 constructor  $\leftarrow$  getConstructorBestArc();
2 s  $\leftarrow$  constructor.getSolution();
3 s  $\leftarrow$  VND(s);
   /* if feasible solution found in at most 3 minutes, break,
   otherwise run in 6 minutes */
4 while keepRunning() do
   | /* During the loop, total arcs required removed from the
   | routes is 50% (this value was chosen because of
   | experiments) */
5   | stmp  $\leftarrow$  destruction(s);
6   | stmp  $\leftarrow$  constructor.completeSolution(stmp.getRoutes());
7   | stmp  $\leftarrow$  VND(stmp);
8   | if s.getValue() > stmp.getValue() then
9   | | s  $\leftarrow$  stmp;
10 return s;

```

---

As buscas locais utilizadas pelo VND são executadas na seguinte ordem: *Reduce*, *OneDistinct*, *Insertion* e *Exchange*.

### 3.4 Resultados e discussão

Os algoritmos foram implementados na linguagem de programação C++, sendo executados em uma *thread* por instância em um cluster que possui 2 processadores Intel(R) Xeon(R) CPU X5650 (12M Cache, 2.66 GHz, 6.40 GT/s Intel(R) QPI, 6 cores, 12 threads), 24 GB de RAM DDR3 1333 MHz, 512 GB de capacidade de armazenamento e 1 porta QDR infiniband de 40 Gbs. Para resolver o modelo matemático, foi utilizado o software CPLEX versão acadêmica 12.5, ajustando o tempo limite para 2 horas com apenas uma *thread* disponibilizada, mantendo como padrão as demais configurações.

As instâncias foram as mesmas utilizadas por Ávila et al. [2017] e estão disponíveis em <https://www.uv.es/corberan/instancias.htm>. São 72 arquivos contendo dados relativos ao grafo e localização dos consumidores. Os parâmetros  $K$  e  $T_{max}$ , que representam a quantidade de veículos e distância máxima percorrida por cada veículo respectivamente, encontram-se juntamente com os resultados de Ávila. A combinação entre os 72 arquivos e os parâmetros  $K$  e  $T_{max}$  totaliza 251 instâncias. Na Tabela 3.3 é disponibilizada uma visão geral dos dados das instâncias, onde as colunas representam: o repositório, total de instâncias, quantidades de vértices, quantidade mínima e máxima de arcos ( $|A|$ ), mínimos e máximos de arcos candidatos ( $|A_r|$ ), mínimos e máximos de arcos que não atendem consumidores ( $|A_{nr}|$ ) e total de consumidores a serem atendidos ( $|H|$ ). Os resultados disponibilizados pelos autores mostraram-se superiores em tempo e qualidade aos obtidos por meio do modelo apresentado na Seção 3.2, por isso os resultados deles foram utilizados na comparação com o IG + VND.

Tabela 3.3: Características das instâncias

Repositório	Total	$ V $	$ A $		$ A_r $		$ A_{nr} $		$ H $	
			Min	Max	Min	Max	Min	Max	Min	Max
Random50	35	50	296	300	105	292	7	193	10	100
Random75	44	75	448	450	143	438	10	305	15	150
Albaida	86	116	259	305	124	172	109	162	18	33
Madrigueras	86	196	453	544	224	305	197	281	22	47

A heurística IG + VND foi executada 10 vezes e, nos testes realizados, um tempo limite de 3 minutos foi suficiente para obtenção de soluções que atendessem todas as restrições para quase todas as instâncias trabalhadas. Para as poucas

instâncias que falharam, foi disponibilizado um tempo extra de mais 3 minutos para que o IG + VND conseguisse encontrar soluções viáveis.

Para melhor organizar esta seção, os dados referentes a execução de cada repositório foram posicionados em tabelas no Apêndice A e apenas as informações a respeito delas foram expostas nas seções a seguir.

Para citar a(s) instância(s) diretamente, seus nomes foram formatados em itálico e a quantidade de veículos em sobrescrito, como no exemplo: *inst*<sup>( $k_1, \dots, k_n$ )</sup>.

Nas figuras apresentadas nas próximas seções, foram descartados os dados em que não foram encontradas soluções viáveis em pelo menos uma das dez execuções feitas pela heurística e, também descartados os dados das instâncias que Ávila não obteve resultado. Foram utilizados 4 algoritmos *Branch-and-Cut* no trabalho Ávila et al. [2017] e, para realizar o comparativo entre tempos de execução com o IG + VND, foi utilizado o algoritmo que apresentou o melhor desempenho.

As seis seções, apresentadas a seguir, são divididas da seguinte forma: na primeira seção, é mostrada a calibração da taxa de destruição utilizada pelo IG + VND; nas quatro seções subsequentes, são expostas as análises dos resultados para cada repositório; na sexta seção, finaliza-se com a análise em conjunto de todos os dados.

### 3.4.1 Calibração da taxa de destruição da heurística

Experimentos estatísticos foram utilizados calibrar para a taxa de destruição utilizada pela heurística IG + VND. Os resultados obtidos pelo experimento foram analisados por intermédio da ferramenta estatística R.

O teste da Análise de Variância (ANOVA) paramétrica foi utilizado para comparar os resultados das taxas de destruição. Os três principais pressupostos do experimento são apresentados abaixo:

**Amostras independentes** Os seguintes itens foram utilizados para a realização do experimento:

- Todas as 251 instâncias.
- 4 configurações de taxa de destruição (20%, 35%, 50% e a taxa variável 20 – 50%<sup>6</sup>).

---

<sup>6</sup>Durante o tempo de execução, a taxa de destruição aumenta enquanto não há melhoria da solução. Quando há melhoria ou a taxa de 50% já foi utilizada em várias iterações, a taxa retorna a 20%.

- 10 execuções para cada par de instância e configuração de taxa de destruição.
- Tempo limitado a 1 minuto de execução.

Através destes parâmetros, foram coletados 10.040 observações independentes (251 x 4 x 10).

**Distribuição normal dos resíduos** Teste de *Shapiro-Wilk* (com grau de confiança definido em 99%):

- O valor obtido para o P-valor foi de 0,5118985.
- De acordo com o teste, a 1% de significância, os resíduos podem ser considerados normais.

**Homogeneidade das variâncias** Teste de *Levene* (com grau de confiança definido em 99%):

- O valor obtido para o P-valor foi de 0,675.
- De acordo com o teste, não há diferença estatisticamente significativa entre os desvios padrões e portanto não se rejeita a hipótese de igualdade das variâncias.

Os três pressupostos da ANOVA foram atingidos e a mesma rejeitou a hipótese nula (P-valor de  $2,5766^{-8}$  é menor que 0,01), de modo que pode-se concluir que existe diferença significativa em pelo menos uma das médias das abordagens analisadas.

A Figura 3.9 mostra o gráfico *boxplot* (diagrama de caixa e bigode), no qual é possível verificar o desempenho geral e a variância dos resultados obtidos pelas taxas de destruição. O resultado do teste de *Scott-Knott* é apresentado na Tabela 3.4.

Como o PRVASP é um problema de minimização, os resultados apresentados mostram que, para o conjunto de instâncias testados, a taxa de destruição a 50% é a indicada para o IG + VND.

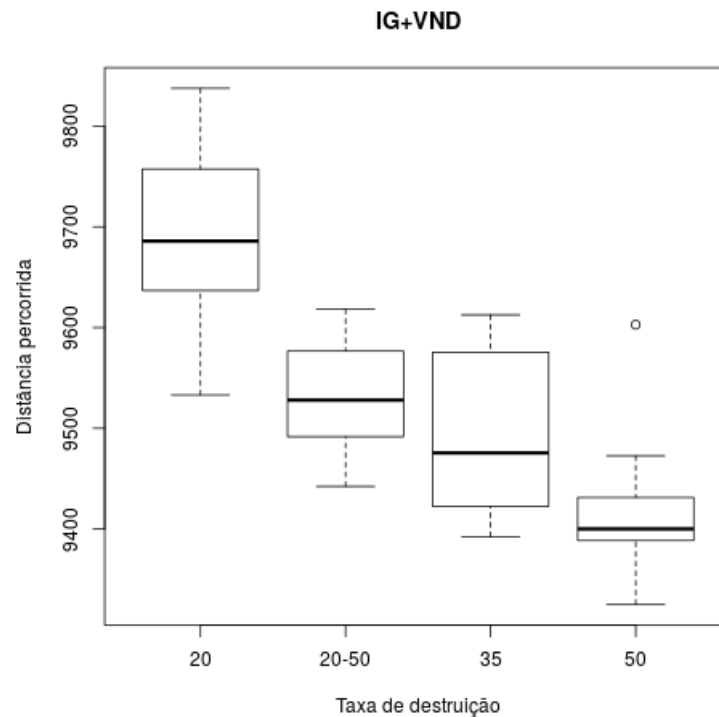


Figura 3.9: Gráfico de médias das distâncias percorridas pelas taxas de destruição utilizadas nos experimentos estatísticos.

Tabela 3.4: Teste de *Scott-Knott*, com nível de confiança de 99%

Grupos	Taxa de destruição	Médias
a	20	9683,727
b	20-50	9531,129
b	35	9487,681
c	50	9418,020

### 3.4.2 Repositório Albaida

O repositório Albaida é composto por 24 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 86 configurações diferentes.

Dos resultados<sup>7</sup> alcançados, foi possível observar que:

- em 75 das 86 instâncias o IG + VND obteve média igual ao resultado de Ávila.
- em nenhuma execução da instância  $A3203\_7^{(5)}$  o IG + VND alcançou o resultado do algoritmo de Ávila.
- todas as soluções viáveis foram encontradas em tempo inferior a 3 minutos.

<sup>7</sup>Os resultados detalhados estão na Tabela A.1 do Apêndice A

- as médias obtidas para os gráficos abaixo vieram de 24, 24, 21 e 16 instâncias, seguindo a ordem de 2, 3, 4 e 5 veículos.

A fim de realizar o comparativo apenas em instâncias em que ambos os métodos de resolução obtiveram soluções viáveis em todas as suas execuções, os dados de uma instância para 5 veículos foi descartada:

- a heurística obteve 1 solução viável em 10 tentativas na instância *A5109\_37*<sup>(5)</sup>.

A Figura 3.10 apresenta a média das médias de custos por quantidade de veículos. Para 2, 3, 4 e 5 veículos, o IG + VND obteve médias 0,06%, 0,05%, 0,02% e 0,22% acima das de Ávila, nesta sequência. A proximidade nos resultados apresentados comprova a qualidade do IG + VND em relação ao método exato nas instâncias deste repositório.

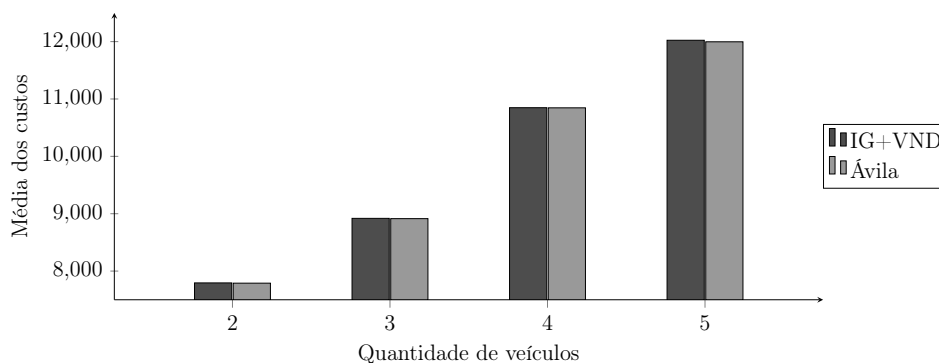


Figura 3.10: Repositório Albaida: gráfico de média de custos por quantidade de veículos.

Na Figura 3.11, os dados de tempo foram obtidos em *Table 4, Algorithm Fxz* Ávila et al. [2017]. O método exato necessitou de mais tempo a cada veículo adicional ao passo que tempo gasto pelo IG + VND manteve-se praticamente constante.

### 3.4.3 Repositório Madrigueras

O repositório *Madrigueras* é composto por 24 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 86 configurações diferentes.

Dos resultados<sup>8</sup> alcançados, foi possível observar que:

- em 36 das 86 instâncias o IG + VND obteve média igual ao resultado de Ávila.

<sup>8</sup>Os resultados detalhados estão na Tabela A.2 do Apêndice A

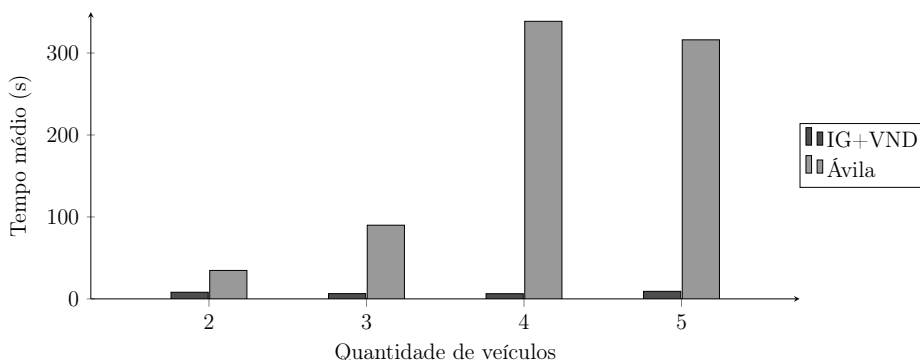


Figura 3.11: Repositório Albaida: gráfico de média dos tempos de execução por quantidade de veículos.

- em nenhuma execução das 5 instâncias,  $M3103\_05^{(3)}$ ,  $M3109\_156^{(2,5)}$ ,  $M3203\_140^{(2)}$  e  $M3205\_55^{(2)}$  o IG + VND alcançou os resultados do algoritmo de Ávila.
- em 11 instâncias,  $M3105\_187^{(3)}$ ,  $M3107\_93^{(5)}$ ,  $M3111\_139^{(4,5)}$ ,  $M3201\_157^{(3,4,5)}$ ,  $M3203\_140^{(4)}$ ,  $M3205\_55^{(4)}$ ,  $M3209\_81^{(5)}$  e  $M3211\_178^{(4)}$  a heurística encontrou melhores resultados do que Ávila.
- todas as soluções viáveis foram encontradas em tempo inferior a 3 minutos.
- as médias obtidas para os gráficos abaixo vieram de 24, 24, 20 e 10 instâncias, seguindo a ordem de 2, 3, 4 e 5 veículos.

A fim de realizar o comparativo apenas em instâncias em que ambos os métodos de resolução obtiveram soluções viáveis em todas as suas execuções, os dados de duas instâncias para 4 e seis para 5 veículos foram descartadas:

- em 8 instâncias,  $M3101\_129^{(4,5)}$ ,  $M3103\_05^{(5)}$ ,  $M3105\_187^{(4,5)}$ ,  $M3203\_140^{(5)}$ ,  $M3205\_55^{(5)}$  e  $M3211\_178^{(5)}$  Ávila não obteve resultados viáveis.

A Figura 3.12 apresenta a média das médias de custos por quantidade de veículos. Para 2 e 3 veículos, o IG + VND obteve médias 0,65% e 0,09% acima das de Ávila, nessa ordem, porém para 4 e 5 veículos a heurística obteve médias 1,92% e 2,14% abaixo. A proximidade nos resultados apresentados (tanto superiores quanto inferiores) comprova a qualidade do IG + VND em relação ao método exato nas instâncias deste repositório, encontrando inclusive resultados melhores para instâncias com mais veículos.

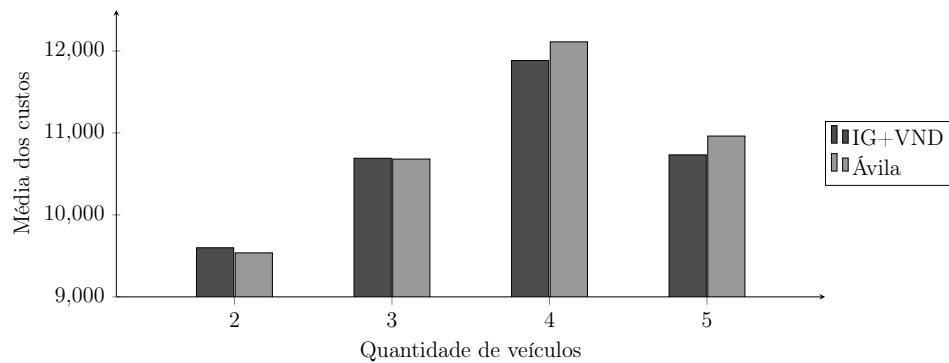


Figura 3.12: Repositório Madrigueras: gráfico de média de custos por quantidade de veículos.

Na Figura 3.13, os dados de tempo foram obtidos em *Table 5, Algorithm Fxz* Ávila et al. [2017]. A cada veículo adicional, o algoritmo de Ávila necessitou de mais tempo ao passo que o tempo gasto pela execução do IG + VND manteve-se praticamente constante.

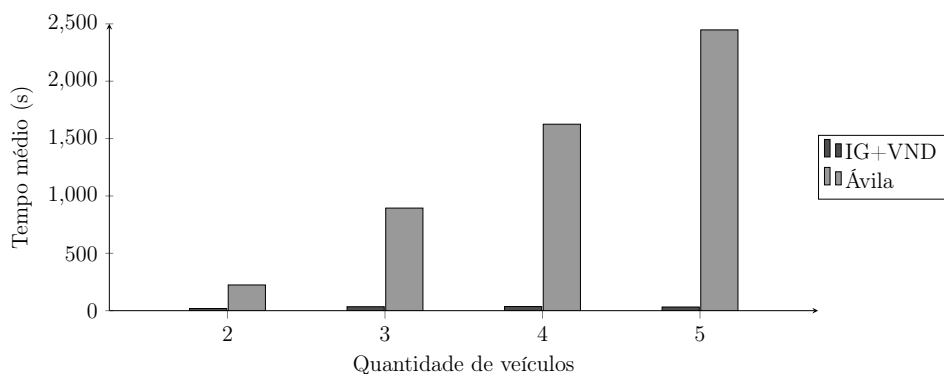


Figura 3.13: Repositório Madrigueras: gráfico de média dos tempos de execução por quantidade de veículos.

### 3.4.4 Repositório Random50

O repositório Random50 é composto por 12 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 35 configurações diferentes.

Dos resultados<sup>9</sup> alcançados, foi possível observar que:

- em 18 das 35 instâncias o IG + VND obteve média igual ao resultado de Ávila.
- em nenhuma execução das 4 instâncias,  $3\_10\_3^{(3)}$ ,  $3\_20\_2^{(2,4)}$  e  $3\_20\_3^{(3)}$  o IG + VND alcançou os resultados do algoritmo de Ávila.

<sup>9</sup>Os resultados detalhados estão na Tabela A.3 do Apêndice A

- na instância  $3\_20\_3^{(5)}$  a heurística encontrou melhor resultado do que Ávila.
- em 2 instâncias,  $3\_10\_2^{(4)}$  e  $3\_20\_3^{(2)}$  houve pelo menos uma execução onde só foram encontradas soluções viáveis em tempo superior a 3 minutos.
- as médias obtidas para os gráficos abaixo vieram de 12, 11, 8 e 3 instâncias, seguindo a ordem de 2, 3, 4 e 5 veículos.

A fim de realizar o comparativo apenas em instâncias em que ambos os métodos de resolução obtiveram soluções viáveis em todas as suas execuções, os dados de uma instância para 4 veículos foi descartada:

- a heurística obteve 9 soluções viáveis em 10 tentativas na instância  $3\_20\_2^{(4)}$ .

A Figura 3.14 apresenta a média das médias de custos por quantidade de veículos. Para 2, 3 e 4 veículos, o IG + VND obteve médias 0,87%, 0,79% e 0,52% acima das de Ávila, respectivamente, porém para 5 veículos a heurística obteve média 0,37% abaixo. A proximidade nos resultados apresentados (tanto superiores quanto no inferior) comprova a qualidade do IG + VND em relação ao método exato nas instâncias deste repositório.

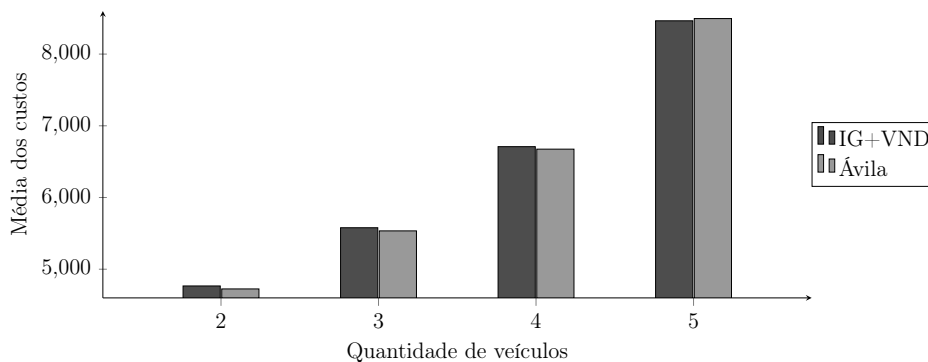


Figura 3.14: Repositório **Random50**: gráfico de média de custos por quantidade de veículos.

Na Figura 3.15, os dados de tempo foram obtidos em *Table 2, Algorithm Fxz* Ávila et al. [2017]. A cada veículo adicional, o algoritmo de Ávila necessitou de mais tempo ao passo que o tempo gasto pela execução do IG + VND aumentou ligeiramente.

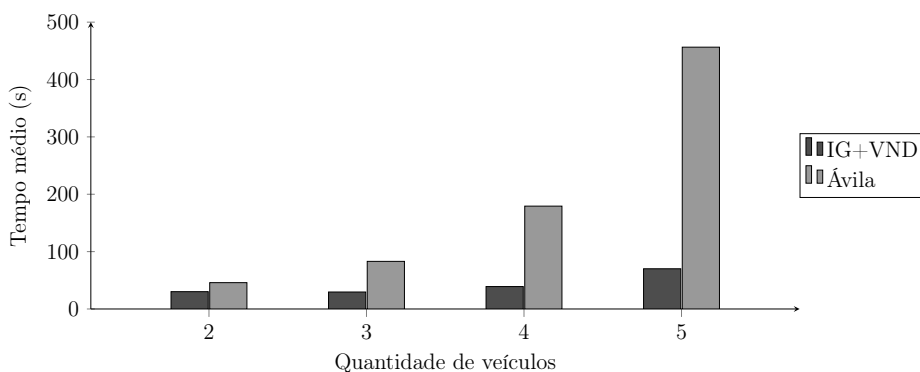


Figura 3.15: Repositório Random50: gráfico de média dos tempos de execução por quantidade de veículos.

### 3.4.5 Repositório Random75

O repositório Random75 é composto por 12 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 44 configurações diferentes.

Dos resultados<sup>10</sup> alcançados, foi possível observar que:

- em 13 das 44 instâncias o IG + VND obteve média igual ao resultado de Ávila.
- em nenhuma execução das 10 instâncias,  $3\_2\_3^{(4,5)}$ ,  $3\_5\_2^{(5)}$ ,  $3\_20\_1^{(2,3)}$ ,  $3\_20\_2^{(2,3)}$  e  $3\_20\_3^{(2,3,4)}$  o IG + VND alcançou os resultados do algoritmo de Ávila.
- em 5 instâncias,  $3\_10\_3^{(5)}$ ,  $3\_20\_1^{(4,5)}$  e  $3\_20\_2^{(5)}$  a heurística encontrou melhores resultados do que Ávila.
- em 10 instâncias,  $3\_10\_3^{(2,3)}$ ,  $3\_20\_1^{(2,4,5)}$ ,  $3\_20\_2^{(2,3,4)}$  e  $3\_20\_3^{(2,3)}$  houve pelo menos uma execução onde só foram encontradas soluções viáveis em tempo superior a 3 minutos.
- as médias obtidas para os gráficos abaixo vieram de 12, 11, 11 e 6 instâncias, seguindo a ordem de 3, 4 e 5 veículos.

A fim de realizar o comparativo apenas em instâncias em que ambos os métodos de resolução obtiveram soluções viáveis em todas as suas execuções, os dados de quatro instâncias (uma para 3 e 4 veículos e duas de 5) foram descartadas:

- a heurística obteve 8, 4 e 9 soluções viáveis em 10 tentativas nas instâncias  $3\_20\_1^{(3,4,5)}$ .

<sup>10</sup>Os resultados detalhados estão na Tabela A.4 do Apêndice A

– na instância  $3\_20\_3^{(5)}$  Ávila não obteve resultado viável.

A Figura 3.16 apresenta a média das médias de custos por quantidade de veículos. Para 2, 3, 4 e 5 veículos, o IG + VND obteve médias 3,56%, 1,70%, 1,10% e 0,87% acima das de Ávila, nesta sequência. O pior resultado divergiu em 3,56%, porém os resultados do IG + VND em relação ao método exato ainda são bem próximos, o que comprova a qualidade da heurística.

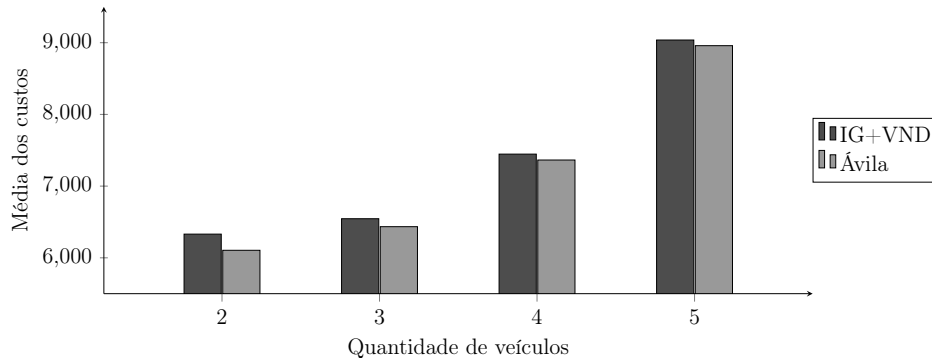


Figura 3.16: Repositório Random75: gráfico de média de custos por quantidade de veículos.

Na Figura 3.17, os dados de tempo foram obtidos em *Table 3, Algorithm Fxz* Ávila et al. [2017]. Como nos repositórios anteriores, a cada veículo adicional, o algoritmo de Ávila necessitou de mais tempo ao passo que o tempo gasto pela execução do IG + VND manteve-se praticamente constante.

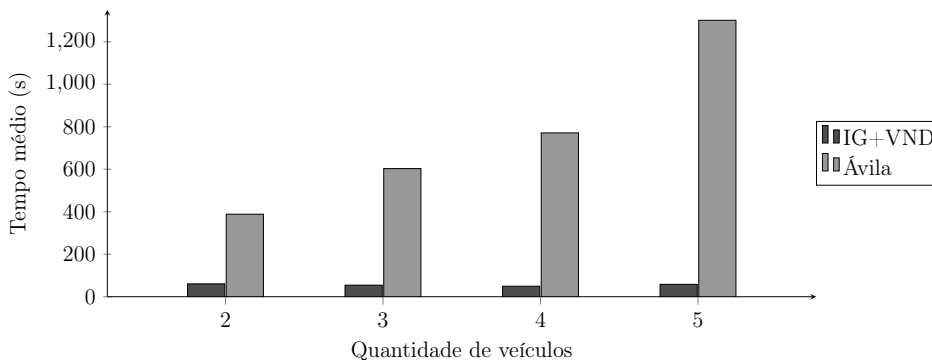


Figura 3.17: Repositório Random75: gráfico de média dos tempos de execução por quantidade de veículos.

### 3.4.6 Geral

Das informações obtidas pelos resultados do Apêndice A, das 251 instâncias testadas podemos observar que:

- em 204 das 221 instâncias onde Ávila garante solução ótima, o IG + VND também conseguiu encontrar o mesmo resultado.
- em 27 instâncias o IG + VND obteve soluções melhores que Ávila pois, em alguns casos não havia solução (8 instâncias) e nos outros, a solução de Ávila não era considerada ótima (19 instâncias).
- em 20 instâncias o IG + VND não conseguiu encontrar soluções iguais às de Ávila.

Comparando os dados de custos por repositório, verificamos que a heurística proposta obteve resultados muito próximos que os de Ávila (0, 10%, 0, 35% e 1, 69% acima nos repositórios Albaida, Random50 e Random75, na devida ordem, e no repositório Madrigueras 0, 90% abaixo), em média, como mostrado na Figura 3.18. Isto comprova que, embora o método heurístico não garanta solução ótima, obteve resultados próximos dentro do conjunto de instâncias analisadas, e em alguns casos melhores, principalmente em instâncias com mais consumidores e veículos.

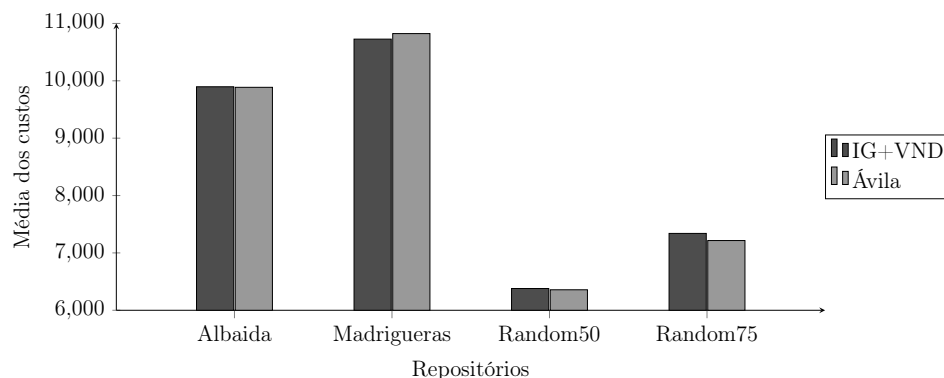


Figura 3.18: Gráfico de média dos custos por repositório.

Métodos heurísticos, desde que bem trabalhados e testados, possuem vantagem de tempo de processamento em relação a métodos exatos e ainda garantem bons resultados. Como pode ser visto na Figura 3.19 e nas seções anteriores, o método exato de Ávila necessitou de mais tempo do que a meta-heurística IG + VND, o que era esperado, sendo que a qualidade dos resultados de ambas as abordagens foram bem próximas.

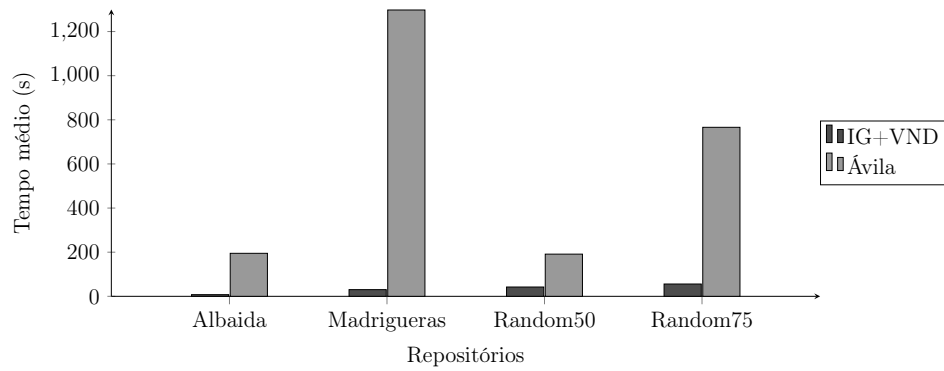


Figura 3.19: Gráfico de tempo médio por repositório.

## Capítulo 4

# Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo

Neste capítulo é proposto o Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), uma derivação do PRVASP. Trata-se de um problema em que ao invés de realizar o atendimento de todos os consumidores como ocorre no PRVASP, visa atender o maior número possível deles. Um caso prático para o problema é quando não é possível atender todos os consumidores em um único dia, seja por causa de redução da frota de veículos por questões mecânicas, seja por redução do número de leituristas por motivo de férias, ou ainda, expansão no número de ruas por causa do crescimento demográfico de uma cidade.

No PRVMASP, com o uso da tecnologia de rádio frequência, os veículos partem do depósito com seus medidores automáticos de leituras instalados, percorrendo as ruas da cidade enquanto realizam a coleta dos dados a distância. Os recursos são limitados, por isso nem sempre é possível atender todos os consumidores, de forma que o objetivo principal é atender a maior quantidade antes de retornar ao depósito.

A definição formal do PRVMASP é mostrada na Seção 4.1. O modelo de programação linear inteira proposto é detalhado na Seção 4.2. Na Seção 4.3 são mostradas a função de avaliação utilizada e as heurísticas desenvolvidas. Por fim, são apresentados os resultados e a discussão sobre eles na Seção 4.4.

## 4.1 Definição do problema

Parecido com o PRVASP, temos um grafo direcionado fortemente conectado  $G = (V, A)$ , onde  $V$  representa o conjunto de vértices (o depósito é o primeiro deles) e, representado as ruas da cidade temos  $A$ , o conjunto de arcos. Associado a cada arco  $(i, j) \in A$  temos um custo  $c_{i,j} \geq 0$  (neste trabalho, o custo representa a distância de travessia entre os vértices  $i$  e  $j$ ).

A frota de  $K$  veículos idênticos é representada pelo conjunto  $\mathbb{K} = \{1, \dots, K\}$  e o conjunto de consumidores é representado por  $\mathbb{W} = \{w_1, \dots, w_W\}$ .  $\mathbb{H} = \{H_1, \dots, H_W\}$  é uma família de subconjuntos de arcos candidatos onde cada  $H_w = \{h_w^i, h_w^{ii}, h_w^{iii}, \dots\}$  é o conjunto de arcos responsáveis pelo atendimento do consumidor  $w \in \mathbb{W}$ . O conjunto de todos os arcos candidatos é dado por  $A_R = H_1 \cup H_2 \cup \dots \cup H_W$  e um arco candidato pode atender a mais de um consumidor, ou seja,  $h = (i, j) \in A_R$  pode pertencer a mais de um  $H \in \mathbb{H}$ .

O PRVMASP tem como objetivo atender a maior quantidade de consumidores à distância, não importando qual veículo o faça, porém para realizar esta tarefa, os veículos precisam partir e retornar ao depósito e também respeitar o limite de percurso  $T_{max}$ .

### 4.1.1 Instância

As instâncias são compostas por:  $K$  veículos; limite de percurso para cada veículo  $T_{max}$ ; grafo  $G = (V, A)$  representando a rede de ruas; custo (distância) de percurso  $c_{i,j} \geq 0$  associado a cada arco; família de subconjunto de arcos candidatos ( $\mathbb{H}$ ) responsáveis pelo atendimento dos consumidores  $w \in \mathbb{W}$ .

Diferentemente do PRVASP que utiliza um pré-processamento na instância, no PRVMASP a instância é utilizada com as informações originais (veja Seção 3.1.1). O PRVMASP tem como objetivo atender o máximo de consumidores e, a ausência de algum consumidor e seus arcos candidatos na execução do método exato ou heurístico traria resultados inconsistentes porque ele não seria contabilizado.

## 4.2 Formulação de programação linear inteira proposta

A formulação matemática do PRVMASP aqui proposta é derivada da formulação apresentada na Seção 3.2 para o PRVASP. Para facilitar o entendimento, as restrições e respectivas explicações são mostradas novamente.

Para ajudar a controlar a passagem dos veículos em cada vértice no modelo (fluxo), denotamos por  $\delta^-(i)$  o conjunto dos arcos que possuem o vértice  $i$  como destino e por  $\delta^+(i)$  o conjunto dos arcos que o possuem como origem, para todo  $i \in V$ . Adicionalmente, para um conjunto  $S \subset V$ , os arcos de saída  $\delta^+(S) = \{(i, j) \in A : i \in S, j \in V \setminus S\}$  junto com os arcos de entrada  $\delta^-(S) = \{(i, j) \in A : i \in V \setminus S, j \in S\}$  formam o conjunto de arcos que começam ou terminam em vértices de  $S$ , ou seja,  $\delta(S) = \delta^+(S) \cup \delta^-(S)$ .

A partir dos dados de entrada da instância, são ajustados os valores de algumas constantes presentes no modelo, sendo elas:

- $M$  - limitante superior para restrições de fluxo, definido como  $|V| - 1$ .
- $M'$  - limitante superior para a quantidade de vezes que um veículo pode passar pelo mesmo arco, sendo seu valor definido por  $\min(M, W + 1, |A_R| + 1)$ .

As variáveis utilizadas são descritas abaixo:

- $w_m$  - variável binária de valor 1 se o consumidor  $m$  foi atendido e 0, caso contrário.
- $y_i^k$  - variável binária que vale 1 se o veículo  $k$  passa pelo vértice  $i$ , 0 caso contrário.
- $x_{i,j}^k$  - variável inteira que contabiliza o número de vezes que o veículo  $k$  passa pelo arco  $(i, j)$ .
- $f_{i,j}^k$  - variável auxiliar, usada para controlar o fluxo do veículo  $k$  no arco  $(i, j)$ .

Em relação ao modelo do PRVASP apresentado na Seção 3.2, são substituídas a função objetivo (3.1) por (4.1), a restrição referente ao atendimento dos consumidores (3.4) por (4.4) e adicionadas as variáveis  $w$  responsáveis por contabilizar os consumidores atendidos. As demais restrições e variáveis permanecem as mesmas. Abaixo o modelo com as alterações citadas:

$$\text{Maximize} \quad \sum_{m=1}^W w_m \quad (4.1)$$

$$\text{s.t.} \quad x^k(\delta^+(0)) \geq 1 \quad \forall k \in \mathbb{K} \quad (4.2)$$

$$x^k(\delta^+(i)) - x^k(\delta^-(i)) = 0 \quad \forall i \in V, \forall k \in \mathbb{K} \quad (4.3)$$

$$\sum_{(i,j) \in H_m} \sum_{k \in \mathbb{K}} x_{i,j}^k \geq w_m \quad \forall H_m \in \mathbb{H}, \forall w_m \in \mathbb{W} \quad (4.4)$$

$$\sum_{(i,j) \in A} c_{i,j} x_{i,j}^k \leq T_{max} \quad \forall k \in \mathbb{K} \quad (4.5)$$

$$\sum_{j:(i,j) \in A} x_{i,j}^k \geq y_i^k \quad \forall i \in V, \forall k \in \mathbb{K} \quad (4.6)$$

$$\sum_{j:(i,j) \in A} x_{i,j}^k \leq M' y_i^k \quad \forall i \in V, \forall k \in \mathbb{K} \quad (4.7)$$

$$f_{i,j}^k \leq M x_{i,j}^k \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (4.8)$$

$$\sum_{(i,j) \in A} f_{i,j}^k \geq \sum_{(i,j) \in A} f_{j,i}^k + 1 - M(1 - y_i^k) \quad \forall i \in V \setminus \{O\}, \forall k \in \mathbb{K} \quad (4.9)$$

$$x_{i,j}^k \in \mathbb{Z}^+ \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (4.10)$$

$$f_{i,j}^k \in \mathbb{R}^+ \quad \forall (i,j) \in A, \forall k \in \mathbb{K} \quad (4.11)$$

$$y_i^k \in \{0, 1\} \quad \forall i \in V, \forall k \in \mathbb{K} \quad (4.12)$$

$$w_m \in \{0, 1\} \quad \forall m \in W \quad (4.13)$$

A equação (4.1) define a função objetivo do problema, a qual busca maximizar a quantidade de consumidores atendidos.

A garantia que o depósito pertença a rota dos veículos é dada pela restrição (4.2) e a restrição (4.3) controla o fluxo de entrada e saída de cada vértice do grafo.

A restrição (4.4) força que se um consumidor é atendido, pelo menos um arco do seu conjunto de arcos candidatos deve pertencer à rota de algum dos veículos.

O limite da distância máxima percorrida por cada veículo fica a cargo da restrição (4.5).

Para garantir que não haja subciclos isolados, temos primeiro as restrições (4.6) e (4.7) que criam uma variável binária garantindo que um dado veículo  $k$  passe por um vértice  $i$ , apenas se o veículo atravessa algum arco incidente a ele. A restrição (4.8) garante a existência de fluxo apenas em arcos utilizados e a restrição (4.9) permite que um arco seja visitado mais de uma vez, formando subciclos presentes na rota, porém impedem a formação de subciclos isolados.

As restrições (4.10), (4.11), (4.12) e (4.13) definem o domínio das variáveis do problema.

## 4.3 Métodos de solução propostos

Nesta seção é apresentada a metodologia utilizada para resolver o PRVMASP de forma heurística. Primeiramente é apresentada a função de avaliação utilizada, após são apresentadas as heurísticas utilizadas na resolução deste problema.

### 4.3.1 Função de avaliação

O objetivo deste problema é atender a maior quantidade de consumidores possível à distância, por intermédio de  $K$  veículos idênticos, respeitando a restrição de limite de percurso  $T_{max}$ .

Diferente do PRVASP, que possui forte dependência da restrição de atendimento dos consumidores, a estratégia utilizada foi a de rejeição de soluções inviáveis, pois a restrição limitante ao problema é o tamanho do percurso e partindo-se de uma solução vazia, já se inicia de uma solução válida.

Uma solução tem melhor valor de função de avaliação se atende mais consumidores ou, caso atenda a mesma quantidade que outra solução, se tem menor distância agregada percorrida pelas  $K$  rotas.

### 4.3.2 Heurísticas

Nesta seção, são discutidas as heurísticas utilizadas para resolver o PRVMASP, passando primeiramente pela construtiva, logo em seguida pelas buscas locais e finalizando com a meta-heurística proposta IG + VND.

#### 4.3.2.1 Heurística construtiva

Para o PRVMASP, foi utilizado praticamente o mesmo algoritmo guloso *BestArc* apresentado na seção 3.3.2.1 do PRVASP. A diferença consiste no fato de não poder ultrapassar o limite permitido para a rota. O efeito desta regra é que, quando um arco candidato não cabe na rota, ele é descartado, independentemente se o consumidor ficar sem atendimento.

#### 4.3.2.2 Busca Local

**Movimentos *Reduce*, *OneDistinct*, *Insertion*, *Exchange*** Estes movimentos, presentes nas buscas locais do PRVASP (Seção 3.3.2.2), foram utilizados nas buscas locais do mesmo modo, com o objetivo de reduzir o custo agregado das rotas, sem que haja redução no número de consumidores atendidos. Um único detalhe foi adicionado: na realização de cada movimento, é descartada a solução cujo movimento ultrapasse o limite da rota, passando a oportunidade para algum outro vizinho ser a solução escolhida.

**Movimento *Replace*** Como no PRVMASP o objetivo é atender a maior quantidade de consumidores, esta busca local surgiu para aproveitar os espaços obtidos pelos outros movimentos, que reduzem o tamanho das rotas, para tentar aumentar o número de consumidores atendidos. Isto é feito substituindo-se um ou mais arcos requeridos por um ou mais arcos candidatos que possam atender mais consumidores.

Para cada rota presente na solução, e ao longo dela, realiza-se um procedimento de duas fases: na primeira fase, de modo semelhante ao apresentado no movimento *Exchange*, remove-se a maior sequência de arcos requeridos até que pelo menos um consumidor não seja mais atendido; na segunda fase, testa a inserção de cada arco candidato, cujo consumidor não era atendido, e procura reduzir ainda mais o custo da rota por caminho mínimo (procedimento semelhante ao do movimento *Reduce*). Nesta etapa, é respeitada a seguinte hierarquia: limite da rota, atendimento do maior número de consumidores e por fim, redução do custo da rota.

Um exemplo do movimento *Replace* pode ser visto na Figura 4.1 e um pseudocódigo no Algoritmo 9.

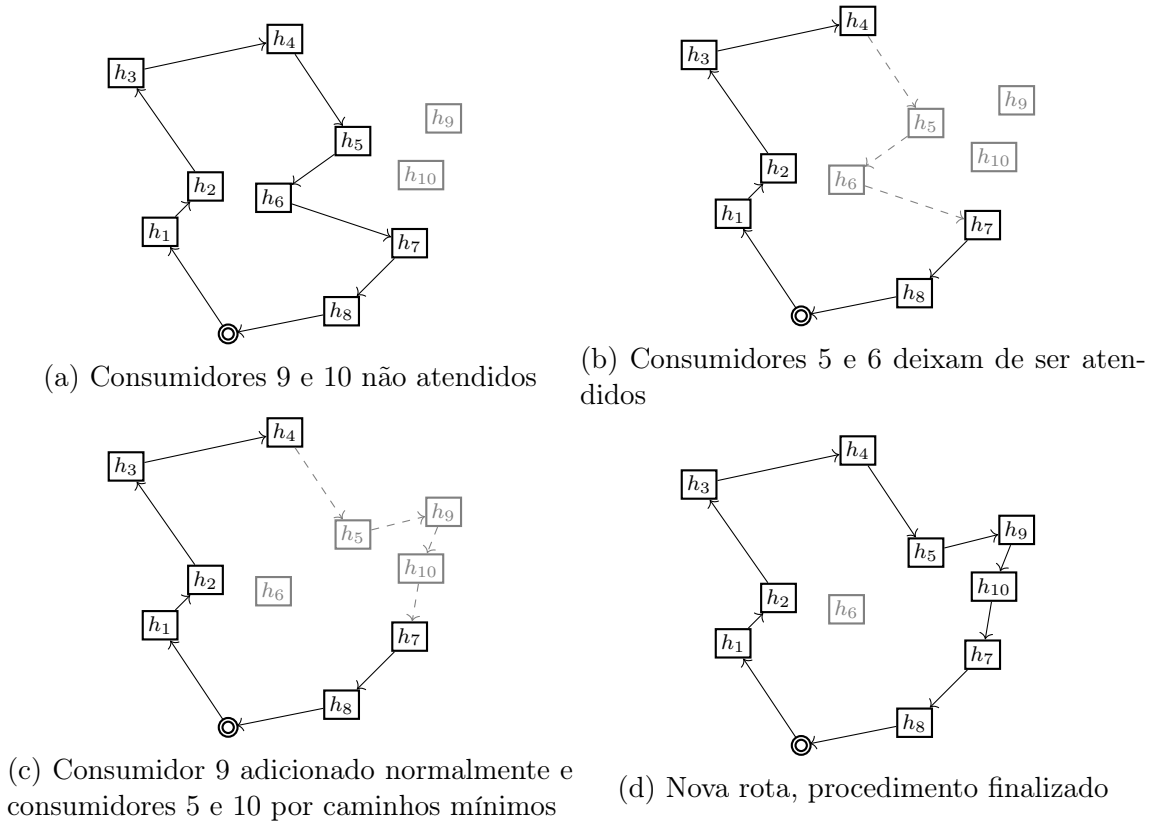


Figura 4.1: Movimento *replace* com auxílio do caminho mínimo para reconexão

---

**Algoritmo 9:** Pseudocode movement *Replace*

---

**Data:**  $graph = (vertices, arcs), s, K$

**Result:** Best solution found

```

1  $s_{tmp} \leftarrow \emptyset;$ 
2  $arcCandidates \leftarrow getAllArcsCostumersUncovered();$ 
3 for  $k = 1$  to  $K$  do
4    $route \leftarrow s.getRoute(k);$            /* required arcs in vehicle  $k$  */
5   for  $h = 1$  to  $arcCandidates.size()$  do
6     for  $i = 1$  to  $route.size()$  do
7       /* Find the biggest  $j$  that can change  $route[i] \rightarrow \dots \rightarrow$ 
8          $route[j]$  to  $route[i] \rightarrow \dots \rightarrow arcCandidates[h] \rightarrow \dots \rightarrow route[j]$ 
9         and increase the costumers covered or keep the same
10        but reduce de route cost (distance) */
11       $j \leftarrow findBiggestJ(i, route, arcCandidates[h]);$ 
12       $route_{aux} \leftarrow createRoute(route, i, arcCandidates[h], j);$ 
13       $s_{aux} \leftarrow (s.getRoutes() - route) \cup route_{aux};$ 
14      if  $s_{aux}.getValue() > s_{tmp}.getValue()$  then
15         $s_{tmp} \leftarrow s_{aux};$ 
16 return  $s_{tmp};$ 

```

---

### 4.3.2.3 Iterated Greedy+VND

A meta-heurística IG + VND utilizada para a resolução do PRVMASP é similar à apresentada na seção 3.3.2.4 do problema PRVASP. As diferenças entre as heurísticas são: vizinhança adicional para o VND (*Replace*) e remoção da condição que estende o tempo de execução acima de 3 minutos, pois sempre haverá solução viável (solução vazia já é uma solução válida).

As buscas locais utilizadas pelo VND são executadas na seguinte ordem: *Reduce*, *OneDistinct*, *Insertion*, *Exchange* e por fim *Replace*.

Um pseudocódigo do IG + VND pode ser visto no Algoritmo 10.

---

#### Algoritmo 10: Pseudocode IG + VND

---

```

Data: graph = (vertices, arcs)
Result: Best solution found
1 constructor  $\leftarrow$  getConstructorBestArc();
2 s  $\leftarrow$  constructor.getSolution();
3 s  $\leftarrow$  VND(s);      /* the sequence neighborhood used in VND is
   Reduce, OneDistinct, Insertion, Exchange and Replace */
4 while keepRunning() do      /* 3 minutes of execution */
   /* During the loop, total arcs required removed from the
   routes is 50% (this value was chosen because of
   experiments) */
5   stmp  $\leftarrow$  destruction(s);
6   stmp  $\leftarrow$  constructor.completeSolution(stmp.getRoutes());
7   stmp  $\leftarrow$  VND(stmp);
8   if s.getValue() < stmp.getValue() then
9     | s  $\leftarrow$  stmp;
10 return s;

```

---

## 4.4 Resultados e discussão

Os algoritmos foram implementados na linguagem de programação C++, sendo o experimento realizado por intermédio de uma *thread* por instância em um cluster que possui 2 processadores Intel(R) Xeon(R) CPU X5650 (12M Cache, 2.66 GHz, 6.40 GT/s Intel(R) QPI, 6 cores, 12 threads), 24 GB de RAM DDR3 1333 MHz, 512 GB de capacidade de armazenamento e 1 porta QDR infiniband de 40 Gbs. Para resolver o modelo matemático, foi utilizado o software CPLEX versão acadêmica 12.5, ajustando o tempo limite para 2 horas com apenas uma *thread* disponibilizada, mantendo como padrão as demais configurações.

Foram utilizadas as mesmas instâncias do PRVASP, mantendo o limite de percurso  $T_{max}$ , porém reduzindo-se em um veículo cada configuração, de modo que a variação do número de veículos passou a ser de 1 a 4. As instâncias juntamente com os parâmetros  $K$  e  $T_{max}$  foram gerados no trabalho de Ávila et al. [2017] de maneira que não fossem inviáveis e nem triviais para o PRVASP, evitando casos em que a solução ótima não utilizasse todos os veículos. Deste modo, a diminuição de veículos das instâncias foi proposta para que não fosse possível o atendimento de todos os consumidores e, como as instâncias já são públicas<sup>1</sup>, é simples a sua reprodução para o PRVMASP.

A heurística IG + VND foi executada 10 vezes com tempo limite de 3 minutos para cada execução. Na Figura 4.2 pode ser observada a evolução da qualidade da solução, em relação a última obtida pela execução, com o passar do tempo pelo IG + VND. Pode-se observar que a convergência ao melhor resultado acontece de forma bem rápida. Em 7 segundos de execução já se alcança 99% do valor da solução final, porém apenas com 174 segundos de execução chega à 100% para todas as execuções de todas as instâncias. Por meio deste gráfico, pode-se verificar a eficiência, em relação ao tempo de execução, do IG + VND no PRVMASP.

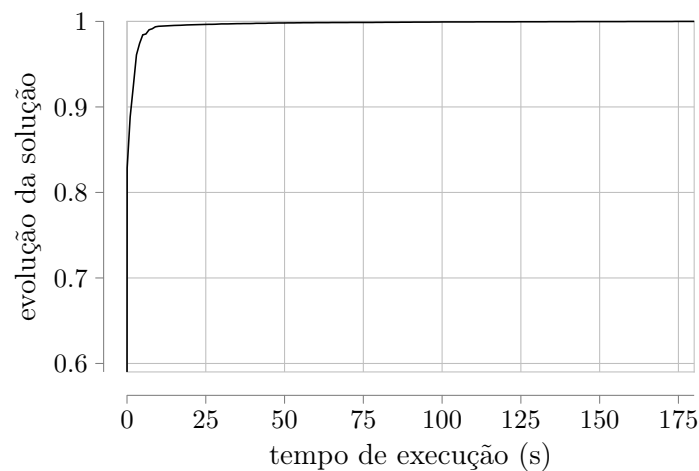


Figura 4.2: Evolução da qualidade da solução, em relação a última obtida pela execução, com o passar do tempo. Informações de tempo coletadas pelas 10 execuções em todas as instâncias.

Para melhor organizar esta seção, os dados referentes a execução de cada repositório para o PRVMASP foram posicionados em tabelas no Apêndice B.

As seis seções a seguir são divididas da seguinte maneira: na primeira é exposta a calibração da taxa de destruição utilizada pelo IG + VND; nas quatro seguintes,

<sup>1</sup><https://www.uv.es/corberan/instancias.htm>

são expostas as análises dos resultados para cada repositório; na sexta seção, finaliza-se com a análise em conjunto de todos os dados. Para os gráficos de tempo, os resultados relativos ao modelo foram omitidos pois, na maioria dos casos, utilizaram todo o tempo de processamento disponibilizado (2 horas). Deste modo, apenas as informações de tempo do IG + VND serão apresentados.

Foram disponibilizadas para os gráficos que contabilizam consumidores as informações a respeito dos limites superiores (*UB - Upper Bound*), que são obtidos mediante a execução final ou parcial (neste problema, interrompida quando excedido o limite de tempo disponibilizado) de cada instância por intermédio do modelo. O valor *UB* garante que não há resultados superiores a ele e, assim, ajuda a validar os resultados obtidos pela heurística.

#### 4.4.1 Calibração da taxa de destruição da heurística

Assim como no PRVASP, também no PRVMASP foram realizados experimentos estatísticos para calibrar a taxa de destruição utilizada pela heurística IG + VND. Os resultados obtidos pelo experimento foram analisados por intermédio da ferramenta estatística R.

O teste da Análise de Variância (ANOVA) paramétrica foi utilizado para comparar os resultados das taxas de destruição. Os três principais pressupostos do experimento são apresentados abaixo:

**Amostras independentes** Os seguintes itens foram utilizados para a realização do experimento:

- Todas as 251 instâncias.
- 4 configurações de taxa de destruição (20%, 35%, 50% e a taxa variável 20 – 50%<sup>2</sup>).
- 10 execuções para cada par de instância e configuração de taxa de destruição.
- Tempo limitado a 1 minuto de execução.

Para estes parâmetros, foram coletados 10.040 observações independentes (251 x 4 x 10).

---

<sup>2</sup>Durante o tempo de execução, a taxa de destruição aumenta enquanto não há melhoria da solução. Quando há melhoria ou a taxa de 50% já foi utilizada em várias iterações, a taxa retorna a 20%.

**Distribuição normal dos resíduos** Teste de *Shapiro-Wilk* (com grau de confiança definido em 99%):

- O valor obtido para o P-valor foi de 0,6404274.
- De acordo com o teste, a 1% de significância, os resíduos podem ser considerados normais.

**Homogeneidade das variâncias** Teste de *Levene* (com grau de confiança definido em 99%):

- O valor obtido para o P-valor foi de 0,002136.
- Como o P-valor foi inferior a 0,01, as variâncias não podem ser consideradas homogêneas.

A ANOVA não pode ser utilizada pois não foram atingidos os três pressupostos esperados (teste de *Levene*). O teste não paramétrico *Kruskal-Wallis* (que funciona de forma análoga à análise ANOVA porém sem as restrições a serem consideradas) foi realizado e obteve P-valor igual a  $2,397^{-6}$ . Com base neste P-valor, fica comprovado pelo teste de *Kruskal-Wallis* que há diferença significativa entre as abordagens utilizadas.

A Figura 4.3 mostra o gráfico *boxplot* (diagrama de caixa e bigode), no qual é possível verificar o desempenho geral e a variância dos resultados obtidos pelas taxas de destruição. O teste de *Scott-Knott* é apresentado na Tabela 4.1.

Como o PRVMASP é um problema de maximização, os resultados apresentados mostram que, para o conjunto de instâncias testados, a taxa de destruição a 50% é a indicada para o IG + VND. Vale ressaltar que a taxa de destruição variável 20 – 50% obteve resultados bem próximos à de 50%.

Tabela 4.1: Teste de *Scott-Knott*, com nível de confiança de 99%

Grupos	Taxa de destruição	Médias
a	50	37,88924
a	20-50	37,87012
b	35	37,83386
c	20	37,72709

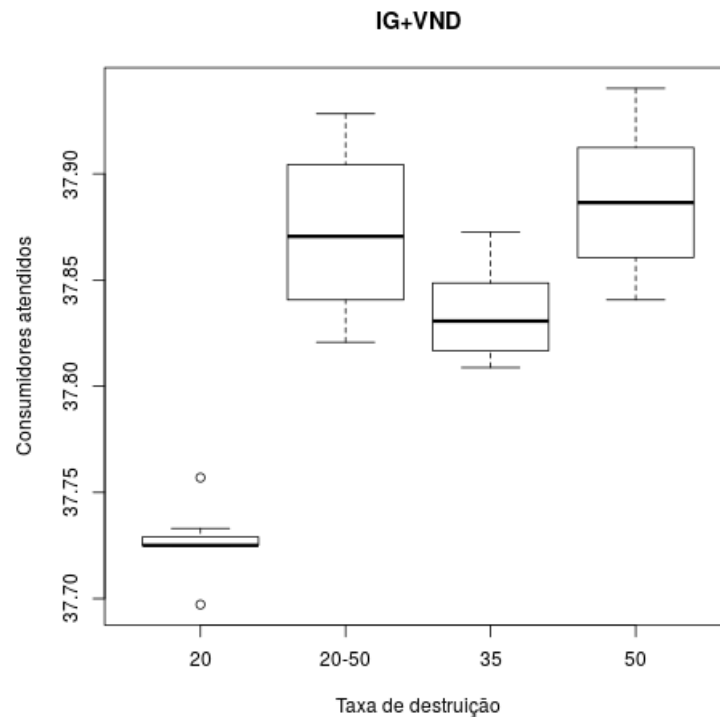


Figura 4.3: Gráfico de médias dos consumidores atendidos pelas taxas de destruição utilizadas nos experimentos estatísticos.

#### 4.4.2 Repositório Albaida

O repositório Albaida é composto por 24 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 86 configurações diferentes.

Dos resultados<sup>3</sup> alcançados, foi possível observar que:

- o modelo encontrou 18 soluções ótimas sendo todas com a configuração de um veículo.
- a divisão das melhores soluções obtidas, nas 86 instâncias, ocorre da seguinte forma:
  - em nenhuma instância o modelo encontrou resultado superior.
  - em 39 instâncias o IG + VND encontrou melhores resultados, sendo para 1, 2, 3 e 4 veículos 2, 10, 15 e 12 instâncias, respectivamente.
  - em 47 instâncias ambas as abordagens encontraram os mesmos resultados. Destas, em 46 o IG + VND obteve média com o mesmo valor do modelo.

<sup>3</sup>Os resultados detalhados estão na Tabela B.1 do Apêndice B

- as médias obtidas para os gráficos abaixo vieram de 24, 24, 21 e 17 instâncias, seguindo a ordem de 1, 2, 3 e 4 veículos.

A Figura 4.4 apresenta a média dos consumidores atendidos por quantidade de veículos. Para 1, 2, 3 e 4 veículos, o IG + VND obteve médias 0,41%, 2,39%, 6,21% e 6,09% acima do modelo, nesta devida ordem. As médias do IG + VND ficaram abaixo do *Upper Bound* 2,06%, 7,72%, 6,51% e 5,94% enquanto as do modelo foram 2,47%, 10,30%, 13,13% e 12,40% inferiores para 1, 2, 3 e 4 veículos, respectivamente. A heurística IG + VND obteve melhores resultados, em relação ao modelo, em todas as quantidades de veículos utilizadas, comprovando assim, sua eficiência neste repositório.

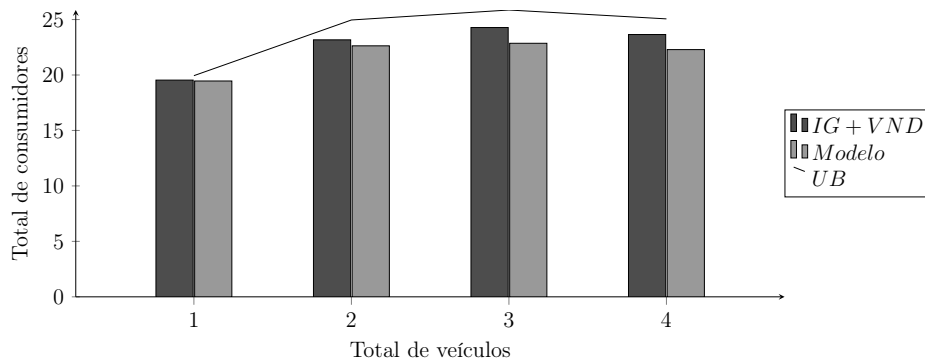


Figura 4.4: Repositório Albaida: gráfico de total de consumidores por número de veículos.

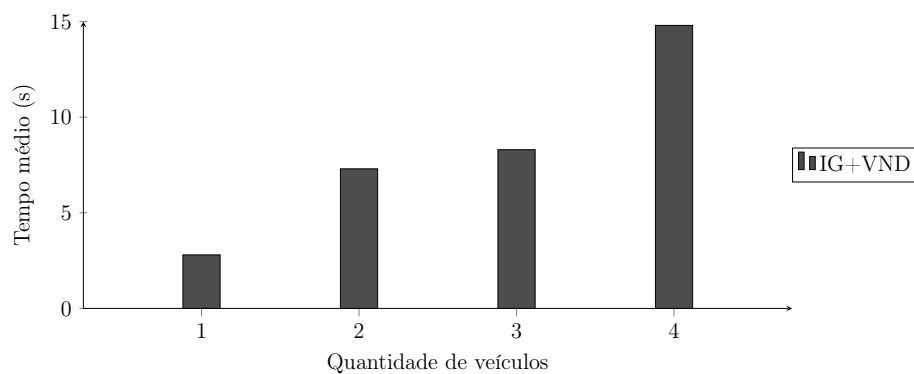


Figura 4.5: Repositório Albaida: gráfico de média dos tempos de execução por quantidade de veículos.

A Figura 4.5 apresenta a média dos tempos por quantidade de veículos. Note que, em média, a convergência ocorreu de maneira muito rápida (pior resultado foi 15 segundos para 4 veículos), o que também garante o uso do IG + VND para o PRVMASP neste repositório.

### 4.4.3 Repositório Madrigueras

O repositório Madrigueras é composto por 24 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 86 configurações diferentes.

Dos resultados<sup>4</sup> alcançados, foi possível observar que:

- o modelo encontrou 8 soluções ótimas sendo todas com a configuração de um veículo.
- a divisão das melhores soluções obtidas, nas 86 instâncias, ocorre da seguinte forma:
  - em nenhuma instância o modelo encontrou resultado superior.
  - em 65 instâncias o IG + VND encontrou melhores resultados, sendo para 1, 2, 3 e 4 veículos 4, 24, 21 e 16 instâncias, respectivamente.
  - em 21 instâncias ambas as abordagens encontraram os mesmos resultados. Destas, em 17 o IG + VND obteve média com o mesmo valor do modelo.
- as médias obtidas para os gráficos abaixo vieram de 24, 24, 22 e 16 instâncias, seguindo a ordem de 1, 2, 3 e 4 veículos.

A Figura 4.6 apresenta a média dos consumidores atendidos por quantidade de veículos. Para 1, 2, 3 e 4 veículos, o IG + VND obteve médias 0,51%, 8,40%, 17,33% e 14,06% acima do modelo, nesta devida ordem. As médias do IG + VND ficaram abaixo do *Upper Bound* 8,13%, 7,52%, 3,39% e 1,62% enquanto as do modelo foram 8,69%, 16,56%, 21,55% e 15,90% inferiores para 1, 2, 3 e 4 veículos, respectivamente. A heurística IG + VND obteve melhores resultados, em relação ao modelo, em todas as quantidades de veículos utilizadas, comprovando assim, sua eficiência neste repositório.

A Figura 4.7 apresenta a média dos tempos por quantidade de veículos. Note que, em média, a convergência ocorreu de maneira muito rápida (pior resultado foi 30 segundos para 3 veículos), o que também garante o uso do IG + VND para o PRVMASP neste repositório.

### 4.4.4 Repositório Random50

O repositório Random50 é composto por 12 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 35 configurações diferentes.

<sup>4</sup>Os resultados detalhados estão na Tabela B.2 do Apêndice B

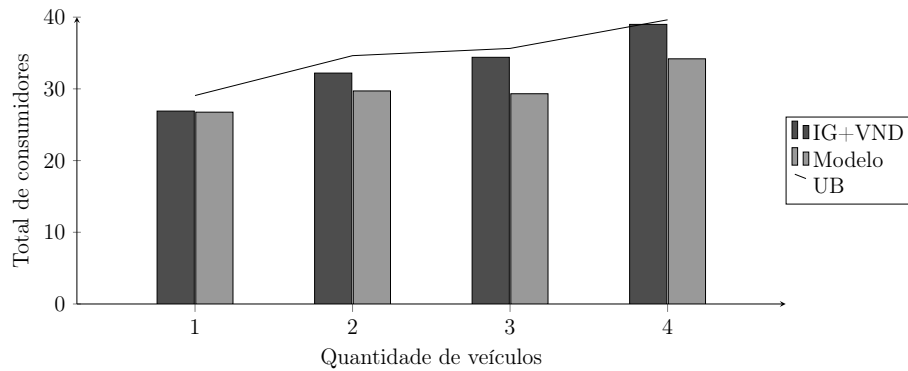


Figura 4.6: Repositório Madrigueras: gráfico de total de consumidores por número de veículos.

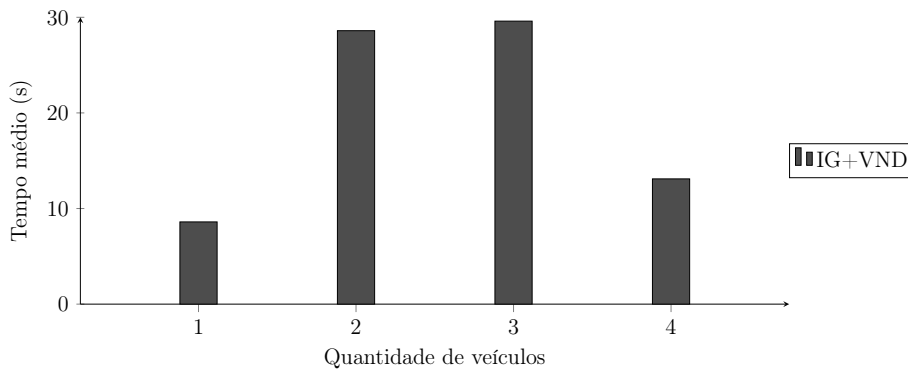


Figura 4.7: Repositório Madrigueras: gráfico de média dos tempos de execução por quantidade de veículos.

Dos resultados<sup>5</sup> alcançados, foi possível observar que:

- o modelo encontrou 7 soluções ótimas sendo todas com a configuração de um veículo.
- a divisão das melhores soluções obtidas, nas 35 instâncias, ocorre da seguinte forma:
  - em 2 instâncias ( $3\_10\_1^{(1)}$  e  $3\_10\_2^{(1)}$ ) o modelo encontrou resultados superiores.
  - em 12 instâncias o IG + VND encontrou melhores resultados, sendo para 2, 3 e 4 veículos 5, 4 e 3 instâncias, respectivamente.
  - em 21 instâncias ambas as abordagens encontraram os mesmos resultados. Destas, em 15 o IG + VND obteve média com o mesmo valor do modelo.

<sup>5</sup>Os resultados detalhados estão na Tabela B.3 do Apêndice B

- as médias obtidas para os gráficos abaixo vieram de 12, 11, 9 e 3 instâncias, seguindo a ordem de 1, 2, 3 e 4 veículos.

A Figura 4.8 apresenta a média dos consumidores atendidos por quantidade de veículos. Para 1 veículo, o modelo obteve média superior em 0,90% e, para 2, 3 e 4 veículos o IG + VND obteve médias 1,21%, 1,91% e 2,07% acima do modelo, nesta devida ordem. As médias do IG + VND ficaram abaixo do *Upper Bound* 3,42%, 3,36%, 2,42% e 1,63% enquanto as do modelo foram 2,48%, 4,61%, 4,37% e 3,73% inferiores para 1, 2, 3 e 4 veículos, respectivamente. Para ambas as abordagens de resolução, os resultados foram parecidos (divergindo no máximo 2,07%), de modo que a utilização da heurística IG + VND é comprovada para este repositório.

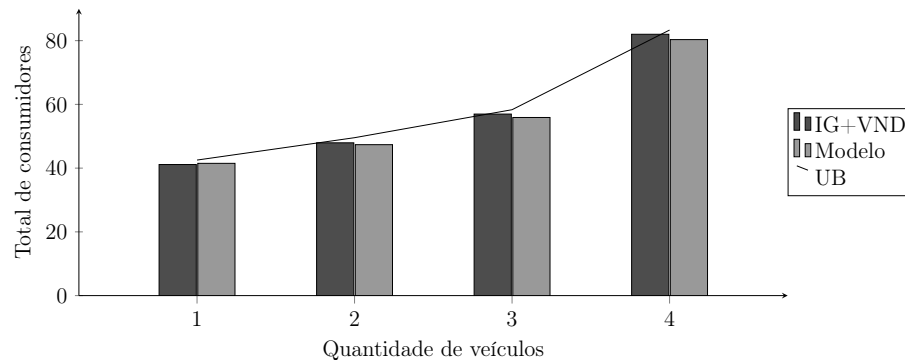


Figura 4.8: Repositório *Random50*: gráfico de total de consumidores por número de veículos.

A Figura 4.9 apresenta a média dos tempos por quantidade de veículos. Note que, em média, a convergência ocorreu de maneira muito rápida (pior resultado foi 63 segundos para 4 veículos), o que também garante o uso do IG + VND para o PRVMASP neste repositório.

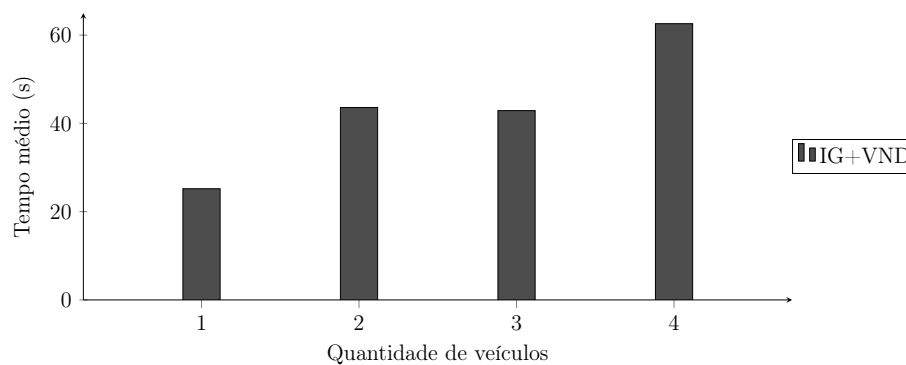


Figura 4.9: Repositório *Random50*: gráfico de média dos tempos de execução por quantidade de veículos.

#### 4.4.5 Repositório Random75

O repositório **Random75** é composto por 12 arquivos, e com as variações do número de veículos ( $K$ ) e distâncias ( $T_{max}$ ) chega-se a 44 configurações diferentes.

Dos resultados<sup>6</sup> alcançados, foi possível observar que:

- o modelo não encontrou soluções ótimas.
- a divisão das melhores soluções obtidas, nas 44 instâncias, ocorre da seguinte forma:
  - em 3 instâncias ( $3\_10\_1^{(1)}$ ,  $3\_10\_2^{(1)}$  e  $3\_20\_2^{(1)}$ ) o modelo encontrou resultados superiores.
  - em 36 instâncias o IG + VND encontrou melhores resultados, sendo para 1, 2, 3 e 4 veículos 6, 11, 12 e 7 instâncias, respectivamente.
  - em 5 instâncias ambas as abordagens encontraram os mesmos resultados. Destas, em 3 o IG + VND obteve média com o mesmo valor do modelo.
- as médias obtidas para os gráficos abaixo vieram de 12, 12, 12 e 8 instâncias, seguindo a ordem de 1, 2, 3 e 4 veículos.

A Figura 4.10 apresenta a média dos consumidores atendidos por quantidade de veículos. Para 1 veículo, o modelo obteve média superior em 0,42% e, para 2, 3 e 4 veículos o IG + VND obteve médias 5,36%, 11,94% e 6,38% acima do modelo, nesta devida ordem. As médias do IG + VND ficaram abaixo do *Upper Bound* 10,73%, 4,73%, 3,68% e 1,85% enquanto as do modelo foram 10,27%, 10,34%, 16,06% e 8,35% inferiores para 1, 2, 3 e 4 veículos, respectivamente. Para um veículo, ambas as abordagens de resolução tiveram resultados parecidos, porém para mais de um veículo, a heurística IG + VND obteve melhores resultados, comprovando assim, sua eficiência neste repositório.

A Figura 4.11 apresenta a média dos tempos por quantidade de veículos. Note que, em média, a convergência ocorreu de maneira muito rápida (pior resultado foi 74 segundos para 4 veículos), o que também garante o uso do IG + VND para o PRVMASP neste repositório. Vale ressaltar que, apesar de bem abaixo do limite imposto de 3 minutos, este é o tempo médio para todas as instâncias por quantidade de veículos. Conforme análise de convergência e evolução da solução apresentada na Figura 4.2, algumas instâncias demandam os 3 minutos, por isto o limite foi mantido neste e nos demais repositórios.

<sup>6</sup>Os resultados detalhados estão na Tabela B.4 do Apêndice B

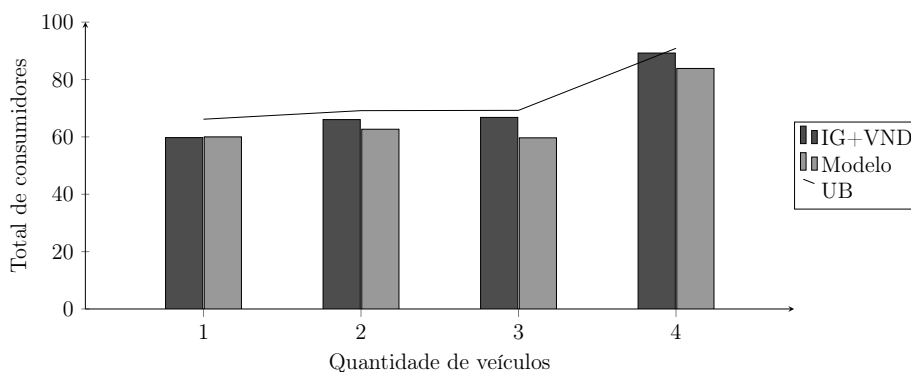


Figura 4.10: Repositório Random75: gráfico de total de consumidores por número de veículos.

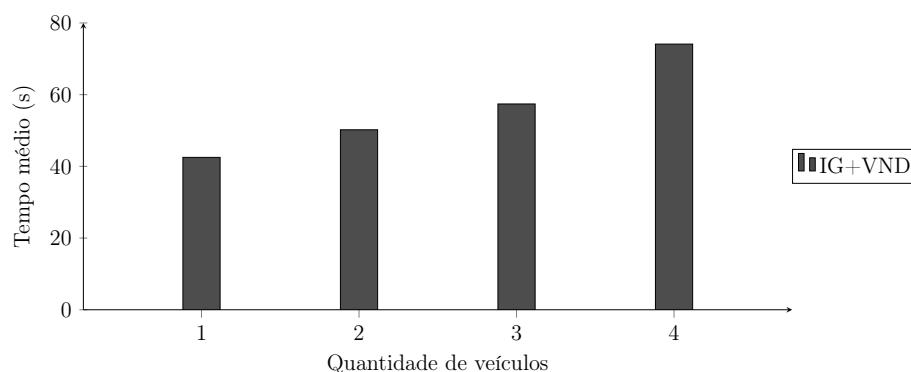


Figura 4.11: Repositório Random75: gráfico de média dos tempos de execução por quantidade de veículos.

#### 4.4.6 Geral

Comparando o dados por repositório, verificamos que a heurística proposta obteve resultados em média melhores de 3,90%, 10,44%, 1,30% e 5,85% nos repositórios *Albaida*, *Madrigueras*, *Random50* e *Random75* em relação modelo proposto, nesta sequência, como mostrado na Figura 4.12. Também é possível observar que os resultados do IG + VND estão próximos ao *Upper Bound* obtido pelo modelo (5,71%, 4,89%, 2,51% e 5,84%). Conclui-se que o resultado da heurística, em média, está comprovadamente abaixo de 6% de distância do ótimo.

Métodos heurísticos não garantam solução ótima porém, desde que bem trabalhados e testados, possuem vantagem de tempo de processamento em relação a métodos exatos e ainda garantem boas soluções. A vantagem em se utilizar o método heurístico, em detrimento ao modelo, na questão do tempo de processamento para o PRVMASP pode ser observada nas figuras 4.12 e 4.13 nas instâncias aqui trabalhadas. Para estas instâncias, por exemplo, limitamos a execução em 3 minutos

o que foi suficiente para encontrar soluções melhores que a do modelo que executou em 2 horas. É reconhecido que o modelo apresentado possa ser melhorado, mesmo assim ele ajudou limitando superiormente os resultados das soluções de maneira a validar e também, qualificar os resultados obtidos por IG + VND.

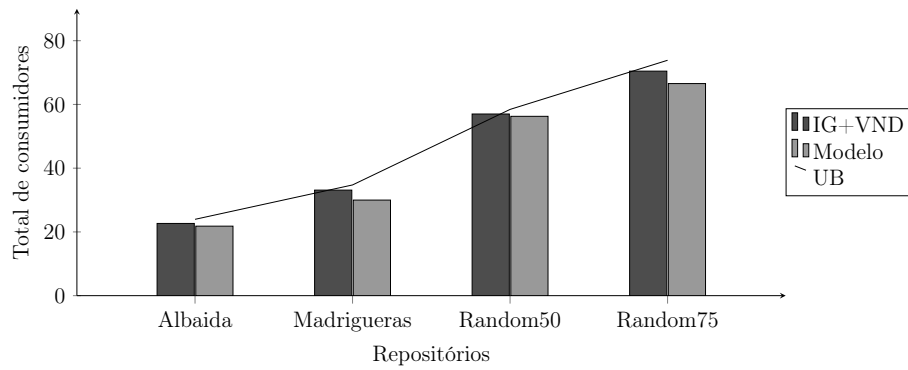


Figura 4.12: Gráfico de média dos custos por repositório.

A Figura 4.13 apresenta a média dos tempos por repositórios.

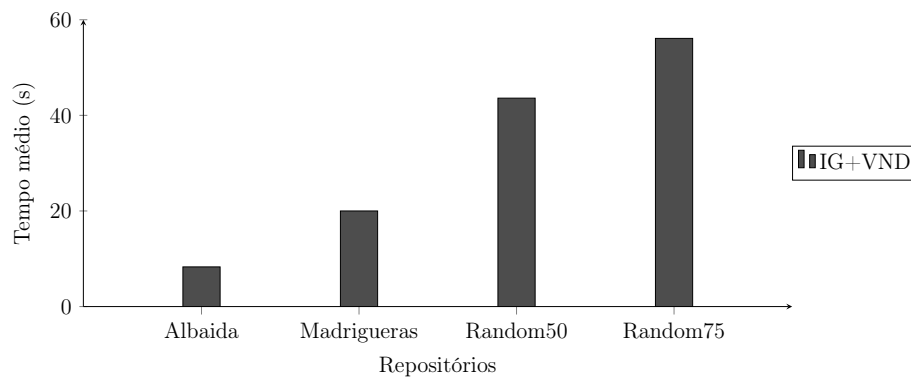


Figura 4.13: Gráfico de tempo médio por repositório.

# Capítulo 5

## Conclusão

Este trabalho abordou dois problemas da área de atendimento de arcos suficientemente próximo. O primeiro problema foi proposto por Ávila et al. [2017], e consiste no atendimento suficientemente próximo com a utilização de múltiplos veículos. Uma nova modelagem foi apresentada neste trabalho, a fim de se obter um melhor entendimento do problema para que fosse possível resolvê-lo de modo heurístico. Para resolver o Problema de Roteamento de Veículos com Atendimento Suficientemente Próximo (PRVASP) de modo heurístico, inicialmente apresentamos técnicas simples para a redução de dados das instâncias, após isso, uma representação de solução que se mostrou mais adequada e, usando esta representação, foi proposta uma meta-heurística híbrida do *Iterated Greedy* com *Variable Neighborhood Descent* (IG + VND), sendo o VND composto pelas buscas locais *Reduce*, *OneDistinct*, *Insertion* e *Exchange*.

As instâncias e os melhores resultados encontrados por quatro algoritmos exatos *Branch-and-Cut*, obtidos por Ávila et al. [2017], foram utilizados na comparação com o IG + VND. A heurística proposta conseguiu resultados equiparáveis aos da literatura na grande maioria dos casos, encontrou soluções viáveis para todos (pior caso foi 1 em 10 tentativas) e encontrou soluções viáveis para instâncias em que não se conhecia nenhuma. Em análises realizadas quando agrupadas as instâncias por veículos, foi observado que o pior desempenho da meta-heurística divergiu 3,56% em relação ao método exato, fato ocorrido no repositório *Random75*, porém quando agrupados por repositórios o pior resultado divergiu em apenas 1,69%. Em todas as análises temporais realizadas, verificou-se a superioridade do IG + VND em relação ao modelo, principalmente a cada veículos a mais utilizado.

No segundo problema, Problema de Roteamento de Veículos com Máximo Atendimento Suficientemente Próximo (PRVMASP), uma variação do PRVASP, a

restrição de atendimento de todos os consumidores passa a ser o objetivo, atender o máximo deles. Para a sua resolução exata, foi proposta uma formulação matemática de programação inteira e, para a sua resolução de modo heurístico, foi adicionada a busca local *Replace* à meta-heurística IG + VND proposta para o PRVASP. O modelo proposto, no geral, não conseguiu obter soluções ótimas, sendo superado em qualidade e tempo pelo IG + VND.

Modelos matemáticos garantem a solução exata, porém a quantidade de dados pode inviabilizar seu uso devido a limitação computacional dos dias atuais. A saída para obtenção de soluções com boa qualidade e em tempo hábil pode vir através de abordagens heurísticas, como demonstrado em vários problemas presentes na literatura. Pelos resultados obtidos, acredita-se que a utilização do IG para explorar e o VND para intensificar a busca por soluções pode servir como boa abordagem para problemas de roteamento de veículos com atendimento suficientemente próximo.

## 5.1 Trabalhos Futuros

Para trabalhos futuros nos problemas PRVASP e PRVMASP, sugere-se a utilização de uma nova meta-heurística que trabalhe com solução única ou um conjunto de soluções que utilize a representação de solução aqui proposta, a fim de explorar outras técnicas de maneira a realizar comparativos identificando os pontos fortes e fracos de cada uma.

Especificamente para o PRVMASP, sugere-se a formulação de um modelo de programação inteira que substitua as restrições de eliminação de subciclos pelas apresentadas nos algoritmos *Branch-and-Cut* no trabalho Ávila et al. [2017].

Assim como ocorreu com o PRVMASP, que tem como objetivo o atendimento da maior quantidade possível de consumidores ao invés do menor percurso possível do PRVASP, outras duas variações no objetivo são propostas:

- minimizar a quantidade de dias necessários para realizar a tarefa de cobertura de todos os consumidores a partir dos recursos disponíveis (sabendo previamente que a quantidade de veículos e/ou o tempo disponível são insuficientes para a realização em um único dia).
- minimizar a quantidade de veículos para realizar a cobertura de todos os consumidores, dada uma quantidade específica de dias que podem ser utilizados para a realização do atendimento.

A fim de aproximar um pouco mais o problema de roteamento de arcos suficientemente próximo à realidade, sugere-se a adição de parâmetros que possibilitem o uso de uma variação, até certo limite, no tamanho permitido da rota. Estendendo um pouco mais o trajeto, possibilita-se que o leitorista realize “horas extras” e aproveite para atender mais consumidores próximos a regiões da rota e, em contrapartida reduza o tamanho das demais. Outra variação interessante é em relação à frota de veículos utilizadas, onde não há a garantia por parte da empresa de possuir veículos idênticos para realização da coleta das informações, pois os veículos podem variar no ano de fabricação, no tipo de terreno que é mais adequado e até no gasto com combustível.

# Apêndice A

## Resultados do PRVASP

Cada tabela presente aqui possui as seguintes colunas: *Nome*, *K* e  $T_{max}$  que representam a identificação da instância juntamente com as configurações de número de veículos e limite de rota; *Cust* e *LB* do agrupamento de Ávila, que são os dados disponibilizados pelos autores onde o primeiro é o custo das rotas e o segundo o *Lower Bound* encontrado quando não se conhece a solução ótima; do agrupamento IG + VND, as informações são referentes às soluções viáveis obtidas onde  $Cust_{avg}$  é o custo médio e  $Cust_{best}$  o melhor,  $t_{max}$  é o maior e  $t_{avg}$  o tempo médio obtido no conjunto das soluções (o tempo coletado vem da última alteração na solução viável de cada execução).

Alguns valores encontram-se em negrito nas tabelas para identificar qual dos métodos obteve vantagem em relação ao custo da solução, se o modelo proposto por Ávila ou se a meta-heurística IG + VND (em pelo menos uma de suas execuções).

Na coluna *Nome* presente na Tabela A.1 referente às instâncias do repositório Albaida, foi removido o sufixo “\_gdrpp”.

Tabela A.1: Resultados das instâncias do repositório Albaida

Nome	<i>K</i>	$T_{max}$	Ávila		IG+VND			
			<i>Cust</i>	<i>LB</i>	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
A3101_15	2	4600	7263	-	7263,0	7263	23,9	7,1
	3	3388	8187	-	8187,0	8187	7,9	3,0
	4	3085	9209	-	9209,0	9209	5,4	2,2
A3103_42	2	4194	6219	-	6227,8	6219	10,6	5,3
	3	3213	6979	-	6979,0	6979	134,4	42,1
	4	2354	7706	-	7706,0	7706	10,1	5,3
A3105_13	2	4666	7345	-	7345,0	7345	127,2	29,0
	3	3422	7573	-	7573,0	7573	11,4	4,4
	4	3010	9498	-	9498,0	9498	22,0	5,6

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
A3107_55	2	2445	3945	-	3945,0	3945	3,0	1,1
	3	1893	4222	-	4278,0	4222	148,3	29,1
	4	1480	4803	-	4803,0	4803	154,8	37,7
	5	1345	5402	-	5402,0	5402	5,2	1,9
A3109_102	2	4960	8261	-	8261,0	8261	0,2	0,1
	3	3578	9324	-	9324,0	9324	0,5	0,3
	4	2887	10588	-	10588,0	10588	3,2	1,5
	5	2541	12024	-	12024,0	12024	11,5	6,3
A3111_41	2	11670	19366	-	19449,7	19366	5,0	3,5
	3	7878	21132	-	21186,2	21132	97,5	25,5
	4	5982	21239	-	21239,0	21239	4,6	1,6
	5	5034	24077	-	24077,0	24077	12,8	5,3
A3201_23	2	4172	6802	-	6802,0	6802	129,6	57,5
	3	3295	7015	-	7015,0	7015	17,2	4,5
	4	2857	8656	-	8656,0	8656	35,3	17,4
	5	2529	10139	-	10139,0	10139	12,2	3,4
A3203_7	2	4235	6994	-	7002,2	6994	4,0	2,0
	3	3456	7267	-	7267,0	7267	66,9	30,0
	4	3067	9845	-	9894,4	9845	168,5	45,4
	5	2873	<b>10612</b>	-	10980,8	10976	148,1	52,0
A3205_107	2	4155	6437	-	6437,0	6437	11,9	4,6
	3	3315	7800	-	7800,0	7800	9,9	3,0
	4	2685	8068	-	8068,0	8068	0,6	0,3
	5	2527	10526	-	10584,9	10526	173,8	72,1
A3207_45	2	2183	3719	-	3720,6	3719	10,2	3,5
	3	1623	4198	-	4198,0	4198	5,6	2,5
	4	1343	4786	-	4786,0	4786	1,7	0,5
	5	1203	5447	-	5447,0	5447	4,3	1,6
A3209_92	2	4610	7896	-	7896,9	7896	179,7	57,2
	3	3287	8410	-	8410,0	8410	1,1	0,5
	4	2626	10014	-	10014,0	10014	25,3	7,3
	5	2295	10683	-	10683,0	10683	2,4	0,9
A3211_101	2	9971	16987	-	16987,0	16987	8,4	2,0
	3	7387	19384	-	19384,0	19384	0,0	0,0
	4	6095	21913	-	21913,0	21913	3,2	1,3
	5	5449	24485	-	24485,0	24485	1,6	1,0
A5101_89	2	3145	4856	-	4856,0	4856	0,4	0,1
	3	2435	5232	-	5232,0	5232	0,4	0,1
	4	1813	6691	-	6691,0	6691	8,1	2,1
	5	1768	6689	-	6689,0	6689	1,3	0,5
A5103_17	2	3773	5799	-	5799,0	5799	0,2	0,1
	3	3015	7670	-	7670,0	7670	1,9	0,7
A5105_22	2	3672	5592	-	5592,0	5592	29,3	8,8
	3	2809	6385	-	6385,0	6385	2,7	0,5
A5107_91	2	1859	3036	-	3036,0	3036	0,8	0,3
	3	1410	3327	-	3327,0	3327	1,6	0,7
	4	1074	3884	-	3884,0	3884	1,2	0,5
	5	1018	4557	-	4557,0	4557	2,2	1,1
A5109_37	2	3475	5544	-	5544,0	5544	0,9	0,3
	3	2748	6025	-	6025,0	6025	0,3	0,2
	4	2383	7285	-	7285,0	7285	0,4	0,2
	5	2109	9967	-	<sup>1</sup> 9967,0	<sup>1</sup> 9967	<sup>1</sup> 73,4	<sup>1</sup> 73,4

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
A5111_52	2	11193	18664	-	18664,0	18664	24,8	7,3
	3	9256	21492	-	21492,0	21492	1,3	0,5
	4	7804	26216	-	26216,0	26216	0,3	0,1
A5201_41	2	2820	4463	-	4463,0	4463	24,3	3,6
	3	2122	5505	-	5509,5	5505	2,9	1,2
	4	1772	5716	-	5716,0	5716	0,0	0,0
	5	1598	6376	-	6376,0	6376	0,0	0,0
A5203_83	2	3365	5111	-	5111,0	5111	4,4	0,8
	3	2856	6042	-	6042,0	6042	8,6	1,5
A5205_29	2	3010	4537	-	4537,0	4537	0,2	0,1
	3	2422	5566	-	5566,0	5566	0,4	0,2
	4	2127	6781	-	6781,0	6781	6,9	1,4
	5	1870	8580	-	8580,0	8580	0,6	0,4
A5207_105	2	1884	2996	-	2996,0	2996	2,1	0,4
	3	1525	3666	-	3666,0	3666	1,5	0,5
	4	1255	4435	-	4435,0	4435	5,0	1,6
	5	1187	5231	-	5231,0	5231	0,8	0,2
A5209_33	2	3803	5836	-	5836,0	5836	0,7	0,3
	3	2972	6737	-	6737,0	6737	2,6	0,8
	4	2349	8380	-	8380,0	8380	0,5	0,2
	5	2245	9679	-	9679,0	9679	0,8	0,3
A5211_113	2	10981	19278	-	19278,0	19278	0,1	0,0
	3	9322	24804	-	24804,0	24804	2,4	0,8
	4	8493	32047	-	32047,0	32047	2,6	0,8
	5	8079	37442	-	37442,0	37442	1,1	0,4

Na coluna *Nome* presente na Tabela A.2 referente às instâncias do repositório Madrigueras, foi removido o termo “\_gdrpp” contido no meio.

Tabela A.2: Resultados das instâncias do repositório Madrigueras

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
M3101_129	2	4685	7499	-	7590,6	7499	163,6	46,4
	3	4080	8115	-	8424,0	8115	178,8	54,3
	4	3174	-	8417,25	8921,4	<b>8919</b>	63,4	15,7
	5	2947	-	8452,23	10275,7	<b>10272</b>	169,2	32,1
M3103_05	2	5066	7982	-	7982,0	7982	160,8	26,4
	3	4243	<b>9052</b>	8587,68	9295,7	9161	179,1	90,7
	4	3832	9628	8640,58	9565,6	9539	171,9	86,0
	5	3490	-	8792,18	9774,0	<b>9774</b>	39,3	16,6
M3105_187	2	4855	7852	-	8089,2	7852	144,7	32,7
	3	4095	9454	8461,53	9106,1	<b>9086</b>	117,5	38,0
	4	3943	-	8381,07	9116,3	<b>9086</b>	179,2	60,1
	5	3640	-	8380,6	9366,8	<b>9320</b>	115,3	69,9
M3107_93	2	2582	4350	-	4350,0	4350	154,2	46,6
	3	1720	4627	-	4627,0	4627	89,2	18,5
	4	1505	4912	-	4912,0	4912	108,7	30,0
	5	1343	5681	4895,54	5115,0	<b>5115</b>	22,9	9,6

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
M3109_156	2	5381	<b>8983</b>	-	9136,0	9136	44,0	12,4
	3	3807	9434	-	9434,0	9434	14,7	6,0
	4	3020	9865	-	9865,0	9865	51,4	15,8
	5	2823	<b>10794</b>	9920,7	10862,1	10808	174,6	59,1
M3111_139	2	15105	25328	-	25385,0	25328	89,1	30,3
	3	10847	26552	-	26561,1	26552	178,7	59,6
	4	8718	30997	27293,1	29440,2	<b>29422</b>	143,3	47,7
	5	8186	29686	27326,5	29458,4	<b>29422</b>	84,1	26,2
M3201_157	2	5564	9032	-	9073,6	9032	82,3	26,0
	3	4083	10317	9399,28	9714,7	<b>9667</b>	177,8	88,2
	4	3712	10451	9691,93	10302,5	<b>10100</b>	166,2	107,0
	5	3434	12224	9739,19	10608,8	<b>10554</b>	156,0	73,0
M3203_140	2	5504	<b>8997</b>	-	9085,9	9063	67,6	12,8
	3	3957	9595	-	9690,6	9595	157,5	53,8
	4	3571	10388	9507,92	10000,7	<b>9997</b>	175,8	71,0
	5	3281	-	9514,6	10472,5	<b>10445</b>	112,5	47,1
M3205_55	2	5680	<b>9180</b>	-	9240,6	9211	159,3	54,4
	3	4124	9742	-	9883,0	9742	176,7	53,8
	4	3734	11526	9700,16	10166,6	<b>10097</b>	146,9	61,9
	5	3443	-	9882,73	10674,1	<b>10638</b>	159,7	42,3
M3207_123	2	3134	5464	-	5548,4	5464	131,7	33,0
	3	2103	5762	-	5806,7	5762	150,6	41,7
	4	1845	6160	-	6161,0	6160	167,8	47,7
	5	1459	6598	-	6598,0	6598	48,3	23,0
M3209_81	2	6139	10069	-	10091,6	10069	70,4	21,1
	3	4264	10577	-	10611,4	10577	173,9	69,1
	4	3326	11305	-	11344,8	11305	178,6	60,5
	5	2857	11933	11555,1	11959,7	<b>11920</b>	172,6	59,0
M3211_178	2	16439	27798	-	28030,7	27798	78,6	26,9
	3	12129	31967	-	32246,0	31967	140,9	77,5
	4	9975	38869	35116,9	37741,6	<b>37421</b>	136,1	52,1
	5	9840	-	32235,8	38267,7	<b>37997</b>	163,5	63,7
M5101_75	2	4139	6870	-	6876,0	6870	6,5	2,3
	3	3101	7692	-	7692,0	7692	12,6	6,1
	4	2581	8023	-	8023,0	8023	7,0	4,6
	5	2322	9732	-	9732,0	9732	3,4	1,6
M5103_03	2	4386	6759	-	6759,0	6759	7,0	3,8
	3	3164	8330	-	8330,0	8330	37,2	10,8
	4	2961	9284	-	9284,0	9284	113,9	51,0
M5105_11	2	4629	7040	-	7040,0	7040	26,9	8,9
	3	3379	8644	-	8644,0	8644	55,3	23,2
	4	3066	10086	-	10086,0	10086	17,1	4,6
M5107_121	2	2377	4012	-	4024,4	4012	110,7	20,8
	3	1770	4610	-	4610,0	4610	19,1	4,3
	4	1466	4867	-	4867,0	4867	2,9	1,8
	5	1314	5810	-	5827,2	5810	140,3	57,7
M5109_10	2	5465	8311	-	8311,7	8311	175,2	25,1
	3	4212	9669	-	9669,0	9669	15,2	6,9
M5111_40	2	14013	21390	-	21390,0	21390	11,2	3,7
	3	11093	23005	-	23005,0	23005	3,5	0,8
	4	8904	27899	-	27899,0	27899	13,5	4,0

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
M5201_147	2	3641	5884	-	5884,0	5884	70,2	12,2
	3	2876	6607	-	6607,0	6607	24,6	11,5
	4	2301	8317	-	8317,0	8317	19,8	6,7
M5203_104	2	3672	5587	-	5587,0	5587	0,0	0,0
	3	2949	6795	-	6795,0	6795	0,6	0,2
	4	2407	8212	-	8212,0	8212	29,0	12,5
	5	2235	9449	-	9449,0	9449	3,7	1,5
M5205_98	2	3334	5330	-	5330,0	5330	0,7	0,4
	3	2551	6044	-	6044,0	6044	62,7	8,1
	4	2062	6736	-	6736,0	6736	11,2	3,9
	5	1915	7717	-	7717,0	7717	23,0	8,2
M5207_08	2	2169	3451	-	3451,0	3451	43,7	6,0
	3	1765	4769	-	4771,0	4769	72,2	29,7
	4	1665	5878	-	5878,0	5878	0,0	0,0
M5209_142	2	4154	6069	-	6471,3	6069	0,3	0,2
	3	3346	6900	-	6916,6	6900	157,6	47,2
	4	2741	8834	8707,52	8866,7	8834	161,2	51,5
M5211_23	2	10110	15652	-	15652,0	15652	2,2	1,4
	3	8001	18099	-	18112,5	18099	119,4	26,8

Na coluna *Nome* presente na Tabela A.3 referente às instâncias do repositório Random50, foi removido o prefixo “LCGDRPP\_50\_”.

Tabela A.3: Resultados das instâncias do repositório Random50

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
3_2_1	2	2294	3521	-	3521,0	3521	2,9	1,7
	3	1901	3942	-	3942,0	3942	0,0	0,0
3_2_2	2	2196	3383	-	3383,0	3383	0,0	0,0
	3	1806	3902	-	3902,0	3902	0,0	0,0
3_2_3	2	1751	2591	-	2591,0	2591	0,0	0,0
3_5_1	2	2551	4054	-	4054,0	4054	55,3	18,7
	3	2023	4665	-	4665,0	4665	46,4	17,8
	4	1627	5290	-	5290,0	5290	0,0	0,0
3_5_2	2	2689	4182	-	4182,0	4182	3,4	1,2
	3	2220	4957	-	4957,0	4957	0,0	0,0
	4	1750	6362	-	6362,0	6362	0,4	0,2
3_5_3	2	2735	4009	-	4009,0	4009	5,1	1,9
	3	1952	4966	-	4966,0	4966	1,9	0,5
	4	1756	5345	-	5345,0	5345	1,5	0,5
3_10_1	2	3499	5256	-	5256,0	5256	53,4	19,8
	3	2492	5945	-	5945,0	5945	27,3	7,6
	4	2198	7054	-	7060,0	7054	164,4	75,4
3_10_2	2	3664	5477	-	5713,4	5477	175,7	45,2
	3	2537	5971	-	6096,1	5971	106,9	10,7
	4	2255	6961	-	7037,0	6961	183,3	64,7
3_10_3	2	2845	4238	-	4238,0	4238	44,1	19,1
	3	2204	<b>4729</b>	-	4843,0	4843	52,8	19,7
	4	1723	5948	-	5948,0	5948	56,1	16,3
	5	1643	6627	-	6649,5	6627	77,2	29,2

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
3_20_1	2	4289	6958	-	7018,5	6958	162,3	76,3
	3	3256	7490	-	7508,6	7490	156,8	88,5
	4	2482	8056	-	8067,8	8056	179,8	78,5
	5	2224	9211	-	9238,4	9211	173,6	73,7
3_20_2	2	3734	<b>5801</b>	-	5897,3	5834	165,0	71,4
	3	2877	6560	-	6642,5	6560	176,3	122,7
	4	2233	<b>7484</b>	-	<sup>9</sup> 7725,9	<sup>9</sup> 7579	<sup>9</sup> 176,1	<sup>9</sup> 77,9
3_20_3	2	4611	7230	-	7334,3	7230	194,8	106,8
	3	3452	<b>7752</b>	-	7894,1	7809	120,9	57,9
	4	2583	8379	-	8562,6	8379	140,0	77,3
	5	2293	9648	8948,25	9504,4	<b>9418</b>	175,6	107,3

Na coluna *Nome* presente na Tabela A.4 referente às instâncias do repositório Random75, foi removido o prefixo “LCGDRPP\_75\_”.

Tabela A.4: Resultados das instâncias do repositório Random75

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
3_2_1	2	2712	3969	-	3969,0	3969	0,0	0,0
	3	2193	4649	-	4649,0	4649	0,0	0,0
	4	1674	6001	-	6037,0	6001	153,0	22,8
3_2_2	2	2649	4141	-	4141,0	4141	0,5	0,2
	3	2139	4804	-	4804,0	4804	0,6	0,2
	4	1756	6005	-	6005,0	6005	0,0	0,0
3_2_3	2	2250	3450	-	3450,0	3450	7,0	2,7
	3	1673	4395	-	4395,0	4395	4,6	2,0
	4	1577	<b>5276</b>	-	5282,0	5279	28,7	7,0
	5	1529	<b>6200</b>	-	6406,0	6406	22,8	5,2
3_5_1	2	3471	5521	-	5521,0	5521	171,1	37,4
	3	2615	5971	-	5978,2	5971	107,5	50,0
	4	1973	6792	-	6792,0	6792	26,4	9,7
3_5_2	2	2898	4625	-	4625,0	4625	80,3	31,5
	3	2323	5371	-	5376,3	5371	101,4	36,6
	4	1892	6891	-	6946,1	6891	157,6	35,5
	5	1820	<b>8028</b>	-	8297,4	8229	176,5	62,1
3_5_3	2	2750	4226	-	4295,0	4226	52,8	14,0
	3	2216	4730	-	4730,0	4730	12,2	3,5
	4	1683	6032	-	6032,0	6032	21,7	9,1
3_10_1	2	3889	6307	-	6423,8	6307	155,7	63,4
	3	2814	6807	-	6867,5	6807	94,2	41,9
	4	2276	7527	-	7560,9	7527	128,1	42,0
	5	2007	8039	-	8182,5	8039	159,3	48,7
3_10_2	2	4953	8333	-	8403,7	8333	175,0	62,5
	3	3451	8757	-	8757,0	8757	145,3	76,2
	4	2700	9323	-	9563,6	9323	175,2	84,2
	5	2325	9898	-	9916,0	9898	53,7	37,2
3_10_3	2	4721	7702	-	8196,2	7702	185,0	101,7
	3	3368	8379	-	8555,4	8379	220,1	128,5
	4	2692	8933	-	9032,6	8933	176,0	95,2
	5	2354	9858	8637,59	9800,6	<b>9699</b>	148,0	73,9

Nome	K	$T_{max}$	Ávila		IG+VND			
			$Cust$	$LB$	$Cust_{avg}$	$Cust_{best}$	$t_{max}(s)$	$t_{avg}(s)$
3_20_1	2	5190	<b>8670</b>	-	9696,0	9149	238,6	138,4
	3	3489	<b>9081</b>	-	<sup>8</sup> 9917,3	<sup>8</sup> 9549	<sup>8</sup> 175,3	<sup>8</sup> 114,9
	4	2638	9952	9332,39	<sup>4</sup> 10052,5	<b>49715</b>	<sup>4</sup> 181,2	<sup>4</sup> 133,8
	5	2585	10615	9255,69	<sup>9</sup> 10991,7	<b>910517</b>	<sup>9</sup> 891,6	<sup>9</sup> 260,2
3_20_2	2	5683	<b>9461</b>	-	9941,2	9618	183,3	138,0
	3	3962	<b>9833</b>	-	10360,3	9927	182,0	131,9
	4	3101	10684	9793,6	10887,2	<b>10449</b>	187,6	138,9
	5	2670	11728	9997,75	11621,9	<b>11561</b>	172,5	122,5
3_20_3	2	4168	<b>6856</b>	-	7305,1	7134	181,1	138,8
	3	2886	<b>7082</b>	-	7526,0	7328	184,2	129,7
	4	2245	<b>7540</b>	-	7770,3	7666	140,3	101,0
	5	1924	-	7421,81	8631,1	<b>8471</b>	177,2	125,7

# Apêndice B

## Resultados do PRVMASP

Cada tabela possui as seguintes colunas: *Nome*, *K* e  $T_{max}$  que representam a identificação da instância juntamente com as configurações de número de veículos e limite de rota; *Value*, *UB*, *t*, *GAP* do agrupamento Modelo são total de consumidores cobertos, *Upper Bound*, tempo utilizado e intervalo médio percentual entre instâncias que não foram resolvidas de forma ideal com a melhor solução viável encontrada, nesta ordem; do agrupamento IG + VND, as informações são referentes aos resultados das soluções onde  $Value_{avg}$  é o total médio de consumidores atendidos e  $Value_{best}$  o máximo de consumidores atendidos,  $t_{max}$  é o maior e  $t_{avg}$  o tempo médio obtido no conjunto das soluções (o tempo coletado vem da última alteração na solução viável de cada execução).

Alguns valores encontram-se em negrito nas tabelas para identificar qual dos métodos obteve vantagem em relação ao total de consumidores atendidos, se o modelo proposto ou se a meta-heurística IG + VND (em pelo menos uma de suas execuções).

Na coluna *Nome* presente na Tabela B.1 referente às instâncias do repositório Albaida, foi removido o sufixo “\_gdrpp”.

Tabela B.1: Resultados das instâncias do repositório Albaida

Nome	<i>K</i>	$T_{max}$	Modelo				IG+VND			
			<i>Value</i>	<i>UB</i>	<i>t</i> (s)	<i>GAP</i> (%)	$Value_{avg}$	$Value_{best}$	$t_{max}(s)$	$t_{avg}(s)$
A3101_15	1	4600	28	-	118	-	28,0	28	1,8	0,9
	2	3388	31	33,00	7195	6,45	32,0	<b>32</b>	2,5	1,3
	3	3085	32	33,00	7196	3,13	32,0	32	18,2	8,6
A3103_42	1	4194	26	-	498	-	25,9	26	1,4	0,4
	2	3213	32	33,00	7195	3,13	32,0	32	3,1	1,6
	3	2354	30	33,00	7195	10,00	32,0	<b>32</b>	2,8	1,2

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	$t_{max}(s)$	$t_{avg}(s)$
A3105_13	1	4666	28	-	780	-	28,0	28	148,0	41,3
	2	3422	31	33,00	7194	6,45	32,0	<b>32</b>	2,2	1,1
	3	3010	28	33,00	7195	17,86	32,0	<b>32</b>	10,6	4,6
A3107_55	1	2445	27	-	13	-	27,0	27	2,5	0,9
	2	1893	31	33,00	7189	6,45	31,0	31	168,7	22,3
	3	1480	29	33,00	7189	13,79	32,0	<b>32</b>	5,4	2,3
	4	1345	31	33,00	7190	6,45	32,0	<b>32</b>	4,2	2,0
A3109_102	1	4960	25	-	11	-	25,0	25	0,5	0,1
	2	3578	30	33,00	7195	10,00	30,0	30	0,7	0,2
	3	2887	29	33,00	7194	13,79	30,0	<b>30</b>	64,9	9,1
	4	2541	28	33,00	7195	17,86	31,0	<b>31</b>	54,1	14,0
A3111_41	1	11670	23	-	60	-	23,0	23	6,8	2,1
	2	7878	28	32,44	7195	15,86	28,0	28	178,1	57,0
	3	5982	27	33,00	7194	22,22	28,8	<b>29</b>	117,0	40,1
	4	5034	30	33,00	7195	10,00	30,1	<b>31</b>	124,2	29,9
A3201_23	1	4172	23	-	809	-	23,0	23	0,6	0,2
	2	3295	26	28,00	7195	7,69	27,0	<b>27</b>	29,4	14,6
	3	2857	25	28,00	7195	12,00	27,0	<b>27</b>	69,4	29,8
	4	2529	23	28,00	7191	21,74	27,0	<b>27</b>	22,8	9,0
A3203_7	1	4235	23	-	1729	-	23,0	23	0,5	0,2
	2	3456	25	28,00	7195	12,00	27,0	<b>27</b>	12,2	4,7
	3	3067	22	28,00	7195	27,27	27,0	<b>27</b>	36,2	15,6
	4	2873	22	28,00	7195	27,27	27,0	<b>27</b>	156,4	85,4
A3205_107	1	4155	23	24,00	7195	4,35	23,0	23	0,6	0,3
	2	3315	25	28,00	7195	12,00	26,0	<b>26</b>	171,1	23,4
	3	2685	26	28,00	7195	7,69	27,0	<b>27</b>	22,2	3,9
	4	2527	26	28,00	7194	7,69	27,0	<b>27</b>	143,8	43,9
A3207_45	1	2183	20	-	12	-	20,0	20	0,1	0,1
	2	1623	24	28,00	7194	16,67	24,0	24	5,0	1,9
	3	1343	25	28,00	7195	12,00	26,0	<b>26</b>	1,5	0,4
	4	1203	25	28,00	7193	12,00	26,0	<b>26</b>	6,1	2,0
A3209_92	1	4610	21	-	13	-	21,0	21	0,8	0,3
	2	3287	25	27,50	7184	9,98	25,0	25	1,7	0,5
	3	2626	23	28,00	7194	21,74	25,0	<b>25</b>	0,6	0,3
	4	2295	26	28,00	7192	7,69	26,0	26	0,5	0,3
A3211_101	1	9971	20	-	11	-	20,0	20	1,2	0,4
	2	7387	24	28,00	7190	16,67	25,0	<b>25</b>	92,3	20,7
	3	6095	22	28,00	7196	27,27	25,0	<b>25</b>	1,7	0,8
	4	5449	24	28,00	7189	16,67	26,0	<b>26</b>	135,5	55,5
A5101_89	1	3145	12	14,91	7195	24,29	12,0	12	8,0	2,8
	2	2435	16	18,00	7195	12,50	16,0	16	0,3	0,2
	3	1813	15	18,00	7195	20,00	16,0	<b>16</b>	1,2	0,6
	4	1768	16	18,00	7194	12,50	17,0	<b>17</b>	13,2	4,1
A5103_17	1	3773	14	16,00	7195	14,29	15,0	<b>15</b>	0,1	0,1
	2	3015	15	18,00	7175	20,00	17,0	<b>17</b>	6,2	1,7
A5105_22	1	3672	14	17,00	7171	21,43	14,0	14	4,5	0,6
	2	2809	15	18,00	7167	20,00	17,0	<b>17</b>	14,3	1,9
A5107_91	1	1859	13	-	27	-	13,0	13	0,1	0,0
	2	1410	15	18,00	7194	20,00	15,0	15	0,0	0,0
	3	1074	16	18,00	7195	12,50	16,0	16	0,0	0,0
	4	1018	16	18,00	7195	12,50	17,0	<b>17</b>	3,8	1,4

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
A5109_37	1	3475	14	-	16	-	14,0	14	0,0	0,0
	2	2748	17	18,00	7195	5,88	17,0	17	3,0	1,1
	3	2383	16	18,00	7194	12,50	17,0	<b>17</b>	9,6	5,1
	4	2109	16	18,00	7195	12,50	16,0	16	0,9	0,2
A5111_52	1	11193	14	-	270	-	14,0	14	0,1	0,0
	2	9256	16	18,00	7195	12,50	17,0	<b>17</b>	1,3	0,4
	3	7804	16	18,00	7195	12,50	17,0	<b>17</b>	15,3	6,8
A5201_41	1	2820	15	-	3470	-	15,0	15	0,1	0,0
	2	2122	18	21,00	7194	16,67	18,0	18	0,0	0,0
	3	1772	20	21,00	7194	5,00	20,0	20	0,1	0,0
	4	1598	20	21,00	7134	5,00	20,0	20	3,7	2,7
A5203_83	1	3365	18	19,00	7194	5,56	18,0	18	0,1	0,0
	2	2856	20	21,00	7195	5,00	20,0	20	0,0	0,0
A5205_29	1	3010	15	16,62	7195	10,82	16,0	<b>16</b>	0,5	0,1
	2	2422	20	21,00	7195	5,00	20,0	20	0,1	0,0
	3	2127	20	21,00	7194	5,00	20,0	20	148,2	37,7
	4	1870	19	21,00	7195	10,53	20,0	<b>20</b>	0,9	0,3
A5207_105	1	1884	17	-	8	-	17,0	17	3,8	1,0
	2	1525	20	21,00	7195	5,00	20,0	20	6,7	1,7
	3	1255	20	21,00	7196	5,00	20,0	20	0,8	0,4
	4	1187	20	21,00	7195	5,00	20,0	20	0,6	0,2
A5209_33	1	3803	17	-	18	-	17,0	17	47,3	15,6
	2	2972	20	21,00	7194	5,00	20,0	20	49,1	18,9
	3	2349	20	21,00	7195	5,00	20,0	20	9,4	3,1
	4	2245	20	21,00	7187	5,00	20,0	20	0,9	0,3
A5211_113	1	10981	17	-	9	-	17,0	17	0,0	0,0
	2	9322	19	21,00	7194	10,53	20,0	<b>20</b>	0,0	0,0
	3	8493	19	21,00	7195	10,53	20,0	<b>20</b>	14,0	3,1
	4	8079	17	21,00	7195	23,53	20,0	<b>20</b>	0,7	0,3

Na coluna *Nome* presente na Tabela B.2 referente às instâncias do repositório Madrigueras, foi removido o termo “\_gdrpp” contido no meio.

Tabela B.2: Resultados das instâncias do repositório Madrigueras

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
M3101_129	1	4685	29	33,51	7188	15,56	31,0	<b>31</b>	54,3	22,4
	2	4080	38	42,00	7194	10,53	41,3	<b>42</b>	156,9	62,7
	3	3174	30	42,00	7194	40,00	42,0	<b>42</b>	2,4	1,1
	4	2947	34	42,00	7196	23,53	42,0	<b>42</b>	1,7	0,8
M3103_05	1	5066	31	34,84	7195	12,38	32,0	<b>32</b>	139,8	25,4
	2	4243	40	42,00	7195	5,00	41,0	<b>41</b>	21,4	12,1
	3	3832	34	42,00	7130	23,53	42,0	<b>42</b>	1,3	0,9
	4	3490	36	42,00	7142	16,67	42,0	<b>42</b>	2,4	1,2
M3105_187	1	4855	31	34,17	7181	10,23	31,0	31	36,6	11,8
	2	4095	39	42,00	7193	7,69	40,0	<b>40</b>	98,8	27,5
	3	3943	41	42,00	7192	2,44	42,0	<b>42</b>	0,3	0,3
	4	3640	41	42,00	7195	2,44	42,0	<b>42</b>	0,9	0,6

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
M3107_93	1	2582	31	-	105	-	31,0	31	4,6	1,7
	2	1720	34	42,00	7195	23,53	36,0	<b>36</b>	73,2	21,1
	3	1505	36	42,00	7196	16,67	40,0	<b>40</b>	22,6	10,2
	4	1343	37	42,00	7196	13,51	42,0	<b>42</b>	78,2	26,1
M3109_156	1	5381	31	-	5126	-	31,0	31	9,6	2,6
	2	3807	36	42,00	7191	16,67	37,0	<b>37</b>	138,2	61,5
	3	3020	36	42,00	7190	16,67	40,0	<b>40</b>	16,4	6,0
	4	2823	37	42,00	7195	13,51	41,7	<b>42</b>	93,2	26,5
M3111_139	1	15105	33	-	1312	-	33,0	33	0,0	0,0
	2	10847	36	42,00	7106	16,67	38,1	<b>39</b>	110,8	51,2
	3	8718	35	42,00	7193	20,00	40,0	<b>40</b>	160,4	56,9
	4	8186	37	42,00	7196	13,51	42,0	<b>42</b>	3,8	1,7
M3201_157	1	5564	38	-	2532	-	37,9	38	148,5	28,5
	2	4083	40	47,00	7195	17,50	45,0	<b>45</b>	61,6	20,4
	3	3712	36	47,00	7195	30,56	47,0	<b>47</b>	133,1	23,6
	4	3434	37	47,00	7196	27,03	47,0	<b>47</b>	1,0	0,7
M3203_140	1	5504	38	39,08	7195	2,84	38,0	38	178,1	44,6
	2	3957	40	47,00	7195	17,50	44,8	<b>45</b>	128,9	75,8
	3	3571	38	47,00	7194	23,68	47,0	<b>47</b>	57,4	15,0
	4	3281	44	47,00	7195	6,82	47,0	<b>47</b>	1,2	0,7
M3205_55	1	5680	38	-	4374	-	37,8	38	22,7	4,1
	2	4124	43	47,00	7194	9,30	45,0	<b>45</b>	155,4	46,1
	3	3734	44	47,00	7195	6,82	47,0	<b>47</b>	97,0	23,1
	4	3443	46	47,00	7196	2,17	47,0	<b>47</b>	1,7	0,9
M3207_123	1	3134	35	-	413	-	35,0	35	41,5	8,0
	2	2103	39	44,06	7194	12,97	41,0	<b>41</b>	29,3	4,3
	3	1845	42	47,00	7196	11,90	44,8	<b>45</b>	76,3	23,4
	4	1459	35	47,00	7196	34,29	44,2	<b>45</b>	62,5	16,1
M3209_81	1	6139	35	-	464	-	35,0	35	1,6	0,6
	2	4264	40	47,00	7134	17,50	42,0	<b>42</b>	84,3	20,1
	3	3326	35	47,00	7161	34,29	44,0	<b>44</b>	175,3	96,9
	4	2857	41	47,00	7158	14,63	45,1	<b>46</b>	179,0	95,9
M3211_178	1	16439	35	36,64	7136	4,69	35,0	35	20,8	4,7
	2	12129	40	47,00	7195	17,50	42,7	<b>43</b>	102,1	62,8
	3	9975	38	47,00	7195	23,68	43,3	<b>44</b>	129,9	59,2
	4	9840	43	47,00	7196	9,30	46,9	<b>47</b>	135,6	28,4
M5101_75	1	4139	21	23,49	7194	11,84	21,4	<b>22</b>	16,2	3,6
	2	3101	24	28,00	7195	16,67	25,0	<b>25</b>	117,0	44,8
	3	2581	21	28,00	7195	33,33	26,9	<b>27</b>	171,5	54,1
	4	2322	24	28,00	7189	16,67	27,0	<b>27</b>	6,8	3,7
M5103_03	1	4386	23	27,18	7194	18,19	23,0	23	6,6	1,8
	2	3164	22	28,00	7195	27,27	26,0	<b>26</b>	12,3	3,2
	3	2961	23	28,00	7194	21,74	27,0	<b>27</b>	104,4	57,5
M5105_11	1	4629	24	27,41	7195	14,21	22,4	24	157,7	25,8
	2	3379	25	28,00	7194	12,00	26,0	<b>26</b>	73,8	24,5
	3	3066	25	28,00	7084	12,00	26,8	<b>27</b>	174,8	106,4
M5107_121	1	2377	21	22,89	7183	9,01	21,0	21	1,6	0,5
	2	1770	23	28,00	7188	21,74	24,0	<b>24</b>	6,1	1,0
	3	1466	18	28,00	7194	55,56	26,0	<b>26</b>	9,2	4,5
	4	1314	19	28,00	7196	47,37	26,0	<b>26</b>	2,0	1,1
M5109_10	1	5465	24	26,00	7196	8,33	24,0	24	90,4	9,8
	2	4212	25	28,00	7195	12,00	27,0	<b>27</b>	41,8	20,1

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
M5111_40	1	14013	23	-	854	-	23,0	23	19,6	6,5
	2	11093	26	28,00	7195	7,69	27,0	<b>27</b>	1,4	0,9
	3	8904	22	28,00	7195	27,27	27,0	<b>27</b>	150,8	59,9
M5201_147	1	3641	17	21,86	7193	28,62	17,0	17	0,4	0,1
	2	2876	14	22,00	7194	57,14	20,0	<b>20</b>	145,9	35,6
	3	2301	16	22,00	7194	37,50	20,0	<b>20</b>	119,4	43,9
M5203_104	1	3672	15	22,00	7193	46,67	17,0	<b>17</b>	0,7	0,2
	2	2949	17	22,00	7195	29,41	21,0	<b>21</b>	9,2	3,5
	3	2407	18	22,00	7180	22,22	21,0	<b>21</b>	2,0	0,9
	4	2235	19	22,00	7195	15,79	21,0	<b>21</b>	5,1	2,9
M5205_98	1	3334	16	21,61	7193	35,09	16,0	16	0,1	0,1
	2	2551	20	22,00	7194	10,00	21,0	<b>21</b>	80,2	23,3
	3	2062	20	22,00	7194	10,00	21,0	<b>21</b>	10,8	4,3
	4	1915	17	22,00	7196	29,41	21,0	<b>21</b>	6,1	2,0
M5207_08	1	2169	18	21,50	7194	19,45	18,0	18	6,9	2,1
	2	1765	16	22,00	7195	37,50	20,0	<b>20</b>	4,0	1,1
	3	1665	16	22,00	7195	37,50	21,0	<b>21</b>	0,0	0,0
M5209_142	1	4154	17	19,58	7195	15,17	16,8	17	1,1	0,3
	2	3346	17	22,00	7195	29,41	21,0	<b>21</b>	155,5	26,8
	3	2741	21	22,00	7194	4,76	21,0	21	18,0	3,6
M5211_23	1	10110	18	22,00	7194	22,22	18,0	18	1,1	0,3
	2	8001	19	22,00	7195	15,79	21,0	<b>21</b>	179,2	35,7

Na coluna *Nome* presente na Tabela B.3 referente às instâncias do repositório Random50, foi removido o prefixo “LCGDRPP\_50\_”.

Tabela B.3: Resultados das instâncias do repositório Random50

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
3_2_1	1	2294	8	-	5752	0,00	8,0	8	0,3	0,2
	2	1901	9	10,00	7194	11,11	9,0	9	5,0	2,3
3_2_2	1	2196	7	9,00	7195	28,57	7,0	7	0,5	0,2
	2	1806	9	10,00	7195	11,11	9,0	9	0,0	0,0
3_2_3	1	1751	7	-	1235	0,00	7,0	7	0,0	0,0
3_5_1	1	2551	21	23,00	7195	9,52	21,0	21	0,2	0,1
	2	2023	24	25,00	7194	4,17	24,0	24	45,0	25,4
	3	1627	23	25,00	7194	8,70	24,0	<b>24</b>	2,3	1,4
3_5_2	1	2689	19	24,00	7081	26,32	19,0	19	0,1	0,1
	2	2220	21	25,00	7194	19,05	23,0	<b>23</b>	178,0	41,6
	3	1750	23	25,00	7194	8,70	24,0	<b>24</b>	0,0	0,0
3_5_3	1	2735	21	-	2663	0,00	21,0	21	1,5	0,5
	2	1952	24	25,00	7196	4,17	24,0	24	1,4	0,7
	3	1756	24	25,00	7195	4,17	24,0	24	1,6	0,7
3_10_1	1	3499	<b>46</b>	47,00	7195	2,17	45,0	45	51,9	15,0
	2	2492	46	50,00	7194	8,70	49,0	<b>49</b>	171,5	64,7
	3	2198	49	50,00	7191	2,04	49,0	49	80,3	22,3
3_10_2	1	3664	<b>43</b>	-	4011	0,00	41,0	41	72,0	28,6
	2	2537	45	50,00	7195	11,11	46,8	<b>48</b>	180,0	92,1
	3	2255	49	50,00	7195	2,04	49,0	49	57,7	10,9

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
3_10_3	1	2845	45	-	3153	0,00	44,8	45	162,0	62,9
	2	2204	49	50,00	7189	2,04	48,3	49	174,7	67,5
	3	1723	48	50,00	7195	4,17	47,5	48	186,4	90,0
	4	1643	46	50,00	7194	8,70	48,9	<b>49</b>	179,4	71,3
3_20_1	1	4289	94	-	3953	0,00	93,9	94	244,3	66,2
	2	3256	98	100,00	7196	2,04	98,0	<b>99</b>	171,0	71,8
	3	2482	97	100,00	7195	3,09	98,1	<b>99</b>	178,2	71,7
	4	2224	97	100,00	7196	3,09	98,1	<b>99</b>	69,9	30,6
3_20_2	1	3734	93	-	2189	0,00	92,4	93	159,7	48,0
	2	2877	98	100,00	7195	2,04	98,0	98	152,2	30,1
	3	2233	91	100,00	7195	9,89	98,6	<b>99</b>	170,3	76,5
3_20_3	1	4611	94	96,35	7033	2,50	93,4	94	161,8	80,8
	2	3452	98	100,00	7000	2,04	98,2	<b>99</b>	174,1	84,0
	3	2583	99	100,00	7049	1,01	98,4	99	179,7	112,5
	4	2293	98	100,00	7196	2,04	99,0	<b>99</b>	169,2	85,8

Na coluna *Nome* presente na Tabela B.4 referente às instâncias do repositório Random75, foi removido o prefixo “LCGDRPP\_75\_”.

Tabela B.4: Resultados das instâncias do repositório Random75

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
3_2_1	1	2712	11	14,07	7194	27,90	12,0	<b>12</b>	0,0	0,0
	2	2193	13	15,00	7193	15,38	14,0	<b>14</b>	3,5	1,8
	3	1674	12	15,00	7193	25,00	13,0	<b>13</b>	0,6	0,2
3_2_2	1	2649	11	14,00	7195	27,27	11,0	11	0,2	0,2
	2	2139	13	15,00	7190	15,38	14,0	<b>14</b>	0,0	0,0
	3	1756	12	15,00	7194	25,00	14,0	<b>14</b>	0,0	0,0
3_2_3	1	2250	10	15,00	7195	50,00	11,0	<b>11</b>	0,2	0,1
	2	1673	11	15,00	7195	36,36	12,0	<b>12</b>	1,7	0,3
	3	1577	12	15,00	7195	25,00	14,0	<b>14</b>	0,9	0,3
	4	1529	14	15,00	7120	7,14	14,0	14	0,1	0,0
3_5_1	1	3471	31	35,96	7137	16,00	31,0	31	0,5	0,4
	2	2615	34	37,00	7070	8,82	35,0	<b>35</b>	112,8	20,4
	3	1973	30	37,00	7195	23,33	35,4	<b>36</b>	159,3	34,3
3_5_2	1	2898	25	37,00	7195	48,00	24,9	<b>26</b>	142,8	21,9
	2	2323	28	37,00	7195	32,14	34,0	<b>34</b>	2,5	1,3
	3	1892	24	37,00	7195	54,17	33,7	<b>34</b>	92,8	34,6
	4	1820	30	37,00	7195	23,33	35,2	<b>36</b>	18,9	6,6
3_5_3	1	2750	28	37,00	7194	32,14	30,0	<b>30</b>	26,8	13,2
	2	2216	34	37,00	7194	8,82	35,0	<b>35</b>	5,1	1,8
	3	1683	29	37,00	7061	27,59	36,0	<b>36</b>	104,1	18,8
3_10_1	1	3889	<b>66</b>	72,04	7196	9,14	65,0	65	172,7	72,7
	2	2814	65	75,00	7195	15,38	72,1	<b>73</b>	77,9	37,8
	3	2276	61	75,00	7195	22,95	73,0	<b>73</b>	165,9	59,3
	4	2007	71	75,00	7196	5,63	74,0	<b>74</b>	147,7	41,8
3_10_2	1	4953	<b>70</b>	71,38	7194	1,97	67,4	68	120,5	45,1
	2	3451	68	75,00	7191	10,29	72,0	<b>72</b>	145,2	77,7
	3	2700	64	75,00	7195	17,19	72,9	<b>73</b>	154,8	82,9
	4	2325	72	75,00	7196	4,17	73,1	<b>74</b>	116,3	67,5

Nome	K	$T_{max}$	Modelo				IG+VND			
			Value	UB	t(s)	GAP(%)	Value <sub>avg</sub>	Value <sub>best</sub>	t <sub>max</sub> (s)	t <sub>avg</sub> (s)
3_10_3	1	4721	66	72,00	7196	9,09	66,9	<b>67</b>	174,8	85,7
	2	3368	70	75,00	7195	7,14	71,6	<b>72</b>	163,4	101,6
	3	2692	69	75,00	7195	8,70	72,8	<b>73</b>	174,0	91,9
	4	2354	72	75,00	7196	4,17	73,7	<b>74</b>	172,9	133,0
3_20_1	1	5190	131	142,64	7196	8,88	134,6	<b>137</b>	173,1	91,3
	2	3489	134	148,76	7195	11,02	143,4	<b>144</b>	178,4	103,9
	3	2638	128	150,00	7195	17,19	145,1	<b>146</b>	175,9	125,9
	4	2585	146	150,00	7196	2,74	148,0	<b>148</b>	150,2	110,2
3_20_2	1	5683	<b>142</b>	144,55	7195	1,79	139,7	141	178,4	98,8
	2	3962	147	150,00	7195	2,04	145,4	147	168,1	132,6
	3	3101	140	150,00	7144	7,14	147,0	<b>147</b>	173,6	130,1
	4	2670	134	150,00	7177	11,94	147,9	<b>148</b>	179,7	120,0
3_20_3	1	4168	129	138,30	7126	7,21	123,5	129	159,9	80,3
	2	2886	135	150,00	7195	11,11	143,8	<b>145</b>	178,8	123,5
	3	2245	135	150,00	7157	11,11	144,6	<b>146</b>	169,2	110,7
	4	1924	132	150,00	7195	13,64	147,9	<b>148</b>	168,6	114,0

# Referências Bibliográficas

- Aneel (2015). Agência nacional de energia elétrica - bandeiras tarifárias. <http://www.aneel.gov.br/bandeiras-tarifarias>. Acesso: 02/05/2018.
- Ávila, T.; Corberán, Á.; Plana, I. & Sanchis, J. M. (2015a). A new branch-and-cut algorithm for the generalized directed rural postman problem. *Transportation Science*, 50(2):750--761.
- Ávila, T.; Corberán, Á.; Plana, I. & Sanchis, J. M. (2015b). The stacker crane problem and the directed general routing problem. *Networks*, 65(1):43--55.
- Ávila, T.; Corberán, Á.; Plana, I. & Sanchis, J. M. (2017). Formulations and exact algorithms for the distance-constrained generalized directed rural postman problem. *EURO Journal on Computational Optimization*, 5(3):339--365.
- Ávila Valverde, T. (2014). *Algunos problemas de rutas por arcos*. PhD thesis, Department d'Estadística i Investigació Operativa, Universitat de València, España.
- Balardino, A. F. et al. (2016). Transporte compartilhado com atendimento suficientemente próximo de passageiros. Master's thesis, Departamento de Informática, Universidade Federal de Viçosa, Brasil.
- Ball, M. O. & Magazine, M. J. (1988). Sequencing of insertions in printed circuit board assembly. *Operations Research*, 36(2):192--201.
- Batista, P. C. d. S. (2013). Medição e controle de consumo de água em instalações prediais. Monografia (Bacharel em Engenharia de Controle e Automação), Universidade Federal de Ouro Preto, Brasil.
- Benavent, E.; Carrota, A.; Corberán, A.; Sanchis, J. M. & Vigo, D. (2007). Lower bounds and heuristics for the windy rural postman problem. *European Journal of Operational Research*, 176(2):855--869.

- Blais, M. & Laporte, G. (2003). Exact solution of the generalized routing problem through graph transformations. *Journal of the Operational Research Society*, 54(8):906--910.
- Carpaneto, G.; Dell'Amico, M. & Toth, P. (1995). Exact solution of large-scale, asymmetric traveling salesman problems. *ACM Transactions on Mathematical Software (TOMS)*, 21(4):394--409.
- Christofides, N.; Campos, V.; Corberán, A. & Mota, E. (1986). An algorithm for the rural postman problem on a directed graph. In *Netflow at pisa*, pp. 155--166. Springer.
- Corberán, Á.; Plana, I. & Sanchis, J. M. (2014). The rural postman problem on directed, mixed, and windy graphs. *Arc Routing: Problems, Methods, and Applications; MOS-SIAM Series on Optimization*, pp. 101--127.
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. (2009). *Introduction to algorithms*. MIT press.
- Drexl, M. (2007). *On Some Generalized Routing Problems*. PhD thesis, Faculty of Business and Economics, RWTH Aachen University, Germany.
- Edmonds, J. (1965). Chinese postmans problem. In *Operations Research*, p. B73. Institute for Operations Research and the Management Sciences.
- Edmonds, J. & Johnson, E. L. (1973). Matching, euler tours and the chinese postman. *Mathematical programming*, 5(1):88--124.
- Eiselt, H. A.; Gendreau, M. & Laporte, G. (1995a). Arc routing problems, part i: The chinese postman problem. *Operations Research*, 43(2):231--242.
- Eiselt, H. A.; Gendreau, M. & Laporte, G. (1995b). Arc routing problems, part ii: The rural postman problem. *Operations research*, 43(3):399--414.
- Euler, L. (1736). Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae scientiarum Petropolitanae*, 8:128--140.
- Gomes, H. C. (2008). Relaxação lagrangeana com fixação de variáveis aplicada ao problema de sequenciamento em uma máquina. Master's thesis, Departamento de Engenharia de Produção, Universidade Federal de Minas Gerais, Brasil.

- Gomes, M. J. N.; Coelho Júnior, W. R.; Palhano, A. W. d. C.; Coutinho, E. F.; Castro, G. A. d.; Gomes, F. J. N.; Barcellos, G. C.; Rezende, B. F. & Pereira, L. W. L. (2009). O problema do carteiro chinês, algoritmos exatos e um ambiente mvi para análise de suas instâncias: sistema xnês. *Pesquisa Operacional*, 29(2):323--363.
- Guan, M. (1984). On the windy postman problem. *Discrete Applied Mathematics*, 9(1):41--46.
- Hà, M. H.; Bostel, N.; Langevin, A. & Rousseau, L.-M. (2012). An exact algorithm for the close enough traveling salesman problem with arc covering constraints. In *1st International Conference on Operations Research and Enterprise Systems (ICORES)*, pp. 233--238.
- Hà, M. H.; Bostel, N.; Langevin, A. & Rousseau, L.-M. (2014). Solving the close-enough arc routing problem. *Networks*, 63(1):107--118.
- Lei nº 7.783, de 28 de junho de 1989 (1989). Dispõe sobre o exercício do direito de greve, define as atividades essenciais, regula o atendimento das necessidades inadiáveis da comunidade, e dá outras providências. [http://www.planalto.gov.br/ccivil\\_03/leis/L7783.htm](http://www.planalto.gov.br/ccivil_03/leis/L7783.htm). Acesso: 30/08/2018.
- Lenstra, J. K. & Kan, A. R. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221--227.
- Martínez, F. J. Z. (2008). Series-parallel graphs are windy postman perfect. *Discrete Mathematics*, 308(8):1366--1374.
- Mei-Ko, K. (1962). Graphic programming using odd or even points. *Chinese Math.*, 1:273--277.
- Mohammaditabar, D. (2012). Chinese postman problem. In *Graph Theory for Operations Research and Management: Applications in Industrial Engineering*, pp. 224--228. IGI Global.
- Mourão, M. C. & Pinto, L. S. (2017). An updated annotated bibliography on arc routing problems. *Networks*, 70(3):144--194.
- Pieterse, V. & Paul, E. (2014). Algorithms and theory of computation handbook, 1999, "chinese postman problem", in dictionary of algorithms and data structures [online]. <https://xlinux.nist.gov/dads/HTML/chinesePostman.html>. Acesso: 25/06/2018.

- Portal Brasil (2017). Agências reguladoras fiscalizam a prestação de serviços públicos. <http://www.brasil.gov.br/governo/2009/11/agencias-reguladoras>. Acesso: 30/08/2018.
- Shuttleworth, R.; Golden, B. L.; Smith, S. & Wasil, E. (2008). Advances in meter reading: Heuristic solution of the close enough traveling salesman problem over a street network. In *The Vehicle Routing Problem: Latest Advances and New Challenges*, pp. 487--501. Springer.
- Stützle, T. & Ruiz, R. (2018). Iterated greedy. *Handbook of Heuristics*, pp. 547--577.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*. John Wiley & Sons.
- Trata Brasil (2016). Instituto trata brasil - principais estatísticas no brasil. <http://www.tratabrasil.org.br/saneamento/principais-estatisticas-no-brasil>. Acesso: 02/05/2018.