Planejamento operacional integrando *cross-docking* e roteamento de veículos para um sistema de distribuição

Eduardo Delcides Bernardes

Tese de Doutorado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Eduardo Delcides Bernardes

Planejamento operacional integrando *cross-docking* e roteamento de veículos para um sistema de distribuição

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Franklina Maria Bragion de Toledo

USP – São Carlos
Dezembro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

B518p Bernardes, Eduardo Delcides
Planejamento operacional integrando cross-
docking e roteamento de veículos para um sistema de
distribuição. / Eduardo Delcides Bernardes;
orientadora Franklina Maria Bragion de Toledo. --
São Carlos, 2019.
159 p.

Tese (Doutorado - Programa de Pós-Graduação em
Ciências de Computação e Matemática Computacional) --
Instituto de Ciências Matemáticas e de Computação,
Universidade de São Paulo, 2019.

1. Rede de varejo. 2. Cross-docking. 3.
Roteamento de veículos. 4. Modelagem matemática. 5.
Métodos heurísticos. I. Toledo, Franklina Maria
Bragion de, orient. II. Título.

Eduardo Delcides Bernardes

**Integration of cross-docking and routing problems for a
distribution system**

Thesis submitted to the Institute of Mathematics
and Computer Sciences – ICMC-USP – in
accordance with the requirements of the Computer
and Mathematical Sciences Graduate Program, for
the degree of Doctor in Science. *FINAL VERSION*

Concentration Area: Computer Science and
Computational Mathematics

Advisor: Profa. Dra. Franklina Maria Bragion
de Toledo

**USP – São Carlos
December 2019**

*Dedico este trabalho à minha mãe (in memoriam),
pois seus ensinamentos e seu exemplo sempre se fizeram presentes,
mesmo tendo partido tão cedo.*

AGRADECIMENTOS

A Deus pela oportunidade de crescimento espiritual e intelectual.

A minha orientadora, Franklina, pelas sábias diretrizes e pela dedicação para a execução deste trabalho. Também pela compreensão, pela amizade e pelo grande exemplo como profissional.

Ao Professor Vinícius Armentano pela valiosa colaboração.

Ao Professor Alysson Costa pela experiência única em Melbourne, pela oportunidade de aprendizado, pela colaboração e pela compreensão.

À Professora Socorro por me iniciar na pesquisa e por acreditar na minha capacidade.

Ao meu pai, Delcides, pela dedicação, pela preocupação e pelo exemplo de ser humano.

Às minhas mães de coração, Nazir e Silvânia (Tata), pelo cuidado e pelo carinho sempre.

Ao meu irmão, Henrique, pela disponibilidade de sempre me ajudar.

À minha prima, Dalva, pelo carinho, pela torcida e por sempre me ajudar.

Aos meus amigos da Bahia. Em especial, a Paulo Galão e a Vinícius Arakawa, irmãos de coração, por serem tão prestativos e especiais na minha vida. A Pryscilla e a Renan pela amizade, pelas ajudas e pela convivência tão saudável.

Ao casal, Jesuelem e Bruno, por serem tão atenciosos e amigos especiais.

A Eliana pela amizade tão valiosa e pela preocupação.

A Deisemara e a Alyne pela amizade e por sempre me motivarem.

Aos meus amigos do Laboratório de Otimização. Em especial, a Pedro Castellucci por ter me recebido em Melbourne e pelas discussões sempre enriquecedoras. Também a Cherri, a Leandro e a Larissa pelas conversas e por tantas ajudas.

Aos professores e funcionários do Instituto de Ciências Matemáticas e de Computação (ICMC/USP).

À Universidade Estadual de Santa Cruz (UESC) pelo afastamento e apoio financeiro.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro concedido sob os processos 2016/20504-1 (bolsa no país), 2017/15029-5 (bolsa BEPE) e 2013/07375-0 (CEPID-CeMEAI).

Enfim, a todos familiares e amigos que colaboraram e torceram por mim. Obrigado!

*“Se o conhecimento pode criar problemas,
não é através da ignorância que podemos solucioná-los.”
(Friedrich Nietzsche)*

RESUMO

EDUARDO, D. B. **Planejamento operacional integrando *cross-docking* e roteamento de veículos para um sistema de distribuição**. 2019. 159 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

A gestão de uma cadeia de suprimentos requer o controle e a integração de diferentes processos entre fornecedores e clientes. *Cross-docking* é uma estratégia de distribuição bastante adotada na prática que considera instalações intermediárias (*cross-docks*) e tem como objetivo reduzir custos logísticos, principalmente, com estoques. De acordo com a literatura, tanto no *cross-docking* quanto na rede da qual faz parte, existem diferentes problemas de decisões que devem ser considerados de forma integrada para a definição de um planejamento de distribuição mais eficiente. Neste trabalho, abordamos os problemas de programação de *cross-dock* e de roteamento de veículos de forma integrada baseados em um sistema de distribuição de uma rede de varejo. No que é de nosso conhecimento, existem apenas dois trabalhos similares ao abordado nesta Tese. Porém, estes consideram produtos perecíveis com características específicas, diferentes do contexto aqui considerado. Para o problema estudado, desenvolvemos dois modelos de programação inteira mista. Além disso, apresentamos também uma análise sobre a relevância da integração dos dois problemas ao comparar com estratégias hierárquicas para resolvê-los. Os resultados mostram que é possível obter um planejamento com menor custo total e menos violações dos prazos das entregas utilizando a estratégia integrada aqui proposta. Para tratar problemas de dimensões maiores, desenvolvemos três métodos heurísticos. Primeiro, apresentamos uma heurística construtiva com a qual é possível obter soluções rapidamente, principalmente para instâncias com maior possibilidade de consolidação de cargas. Em seguida, desenvolvemos uma meta-heurística de busca em vizinhança variável com a qual obtivemos soluções com ganhos de até 27% em relação a soluções obtidas para o problema utilizando modelagem e o *solver* Gurobi. Finalmente, adaptamos a matheurística *Proximity Search* para resolver o problema estudado. Neste caso, como esperado, o método apresenta um melhor desempenho para as instâncias maiores.

Palavras-chave: Rede de varejo, *Cross-docking*, Roteamento de veículos, Modelagem matemática, Métodos heurísticos.

ABSTRACT

EDUARDO, D. B. **Integration of cross-docking and routing problems for a distribution system**. 2019. 159 p. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Supply chain management entails controlling and integrating different processes between suppliers and consumers. Cross-docking is a widely adopted distribution strategy which considers intermediate installations (cross-docks) and contributes to reducing logistics costs, in particular, storage costs. According to literature, both in cross-docking and in its network, many decision problems may be integrated to define a more efficient distribution plan. In this study, cross-dock schedule and vehicle routing problems are approached in an integrated way based on a typical retail network. To the best of our knowledge, there are only two papers similar to this study; however, they focus on a distribution system of perishable products with features that are not suitable here. For the problem studied, we develop two mixed integer models and compare the integrated problem solutions with hierarchical strategies solutions through computational experiments. The results show that, with the integrated strategy, it is possible to obtain a lower cost distribution plan and fewer violations of delivery time windows. To deal with larger instances, we propose three solution heuristic methods. First, we present a constructive heuristic that provides a quick solution, mainly to the instances with higher chances of load consolidation. After that, we develop a variable neighborhood search meta-heuristic from which we obtain solutions up to 27% better than the ones obtained using the Gurobi solver. Finally, we adapt the matheuristic Proximity Search to solve the problem. In this case, as expected, the method presents a better performance to larger instances.

Keywords: Retail network, Cross-docking, Vehicle routing, Mathematical modelling, Heuristic methods.

LISTA DE ILUSTRAÇÕES

Figura 1 – Ilustração das três estratégias de distribuição.	23
Figura 2 – Representação de um <i>cross-dock</i> e das atividades nele realizadas.	27
Figura 3 – Representação da estrutura dos modelos que abordam o processamento de cargas de entrada e de saída de forma simultânea.	36
Figura 4 – Ilustrações das duas abordagens do problema integrado de roteamento e <i>cross-docking</i>	42
Figura 5 – Conjuntos de cargas de entrada e de clientes e seus pedidos.	50
Figura 6 – Representação de uma estrutura interna do <i>cross-dock</i> com uma ordenação e consolidação de cargas definidas.	51
Figura 7 – Representação da interdependência entre as decisões para obter um plano de distribuição.	52
Figura 8 – Representação da estrutura dos modelos desenvolvidos para o VRCDSP. . .	52
Figura 9 – Ilustração do funcionamento de um <i>cross-dock</i> em um determinado instante. .	58
Figura 10 – Relaxação linear ($M1 \times M2$).	65
Figura 11 – Número de soluções ótimas obtidas no decorrer de 2 horas ($M1 \times M2$) - 15 clientes.	66
Figura 12 – Limitantes duais e primais obtidos em 2 horas ($M1 \times M2$) - 25 clientes. . .	67
Figura 13 – Comparação entre as quatro estratégias para as instâncias com 15 clientes. .	70
Figura 14 – Tempos de acordo com as estratégias EI e RCD - Cenário 2.	72
Figura 15 – Exemplo de soluções vizinhas de acordo com a vizinhança N_1	94
Figura 16 – Exemplo de soluções vizinhas de acordo com a vizinhança N_2	95
Figura 17 – Exemplo de soluções vizinhas de acordo com a vizinhança N_3	95
Figura 18 – Exemplo de soluções vizinhas de acordo com a vizinhança N_4	96
Figura 19 – Exemplo de soluções vizinhas de acordo com a vizinhança N_5^a	97
Figura 20 – Possibilidades de escolhas para as buscas locais com a vizinhança N_5^b	98
Figura 21 – Possibilidades de escolhas para as buscas locais com a vizinhança N_5^c	98
Figura 22 – Tempos de acordo com as estratégias EI e RCD - Cenário 1.	146

LISTA DE TABELAS

Tabela 1 – Resumo dos elementos e dos objetivos apresentados em Ladier e Alpan (2016).	30
Tabela 2 – Objetivos e características operacionais considerados em cada artigo revisado.	32
Tabela 3 – Objetivos e características operacionais considerados em cada artigo revisado.	34
Tabela 4 – Diferenças entre a literatura e o problema estudado.	45
Tabela 5 – Classificação dos trabalhos de acordo com a estratégia de modelagem separados pela quantidade de docas.	46
Tabela 6 – Parâmetros de cada conjunto de instâncias de acordo com o número de clientes.	63
Tabela 7 – Dados gerados aleatoriamente.	64
Tabela 8 – Dados calculados.	64
Tabela 9 – Comparação dos modelos adaptados.	68
Tabela 10 – Performance da estratégia EI em relação à RCD - 15 clientes.	71
Tabela 11 – Performance da estratégia EI em relação à RCD - 25 clientes.	74
Tabela 12 – Performance da estratégia RCD em relação à EI - 25 clientes.	74
Tabela 13 – Tempo médio de solução em segundos para obter as melhores soluções com as estratégias EI e RCD.	75
Tabela 14 – Performance das versões da heurística em relação ao <i>solver</i> - Cenário 1. . .	85
Tabela 15 – Performance das versões da heurística em relação ao <i>solver</i> - Cenário 2. . .	86
Tabela 16 – Tempos de solução: heurística $\times$ <i>solver</i> (em segundos).	87
Tabela 17 – Desempenho da duas versões da FO-VND - Cenário 1.	100
Tabela 18 – Desempenho da duas versões da FO-VND - Cenário 2.	100
Tabela 19 – Comparação entre a solução inicial (heurística construtiva) com a melhor solução obtida pela meta-heurística.	101
Tabela 20 – Médias dos tempos de solução (em segundos).	102
Tabela 21 – Frequência média de melhorias de acordo com as vizinhanças - Cenário 1. .	103
Tabela 22 – Frequência média de melhorias de acordo com as vizinhanças - Cenário 2. .	103
Tabela 23 – Resultados para algumas combinações das vizinhanças - Cenário 1.	104
Tabela 24 – Resultados para algumas combinações das vizinhanças - Cenário 2.	105
Tabela 25 – Resultados para as instâncias do Cenário 1.	115
Tabela 26 – Resultados para as instâncias do Cenário 2.	116
Tabela 27 – Comparação entre as soluções obtidas com a PS1 e PS2.	117
Tabela 28 – Comparação entre a solução inicial (heurística construtiva) com a melhor solução obtida pela PS2.	117
Tabela 29 – Soluções obtidas com a relaxação linear dos Modelos M1 e M2 - Cenário 1.	130

Tabela 30 – Soluções obtidas com a relaxação linear dos Modelos M1 e M2 - Cenário 2.	131
Tabela 31 – Informações sobre as soluções obtidas com os dois modelos desenvolvido - Cenário 1.	132
Tabela 32 – Informações sobre as soluções obtidas com os dois modelos desenvolvido - Cenário 2.	133
Tabela 33 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 15 clientes.	134
Tabela 34 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 25 clientes.	135
Tabela 35 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 40 clientes.	136
Tabela 36 – Informações sobre as soluções obtidas com a estratégia RCD - 15 clientes. .	137
Tabela 37 – Informações sobre as soluções obtidas com a estratégia RCD - 25 clientes. .	138
Tabela 38 – Informações sobre as soluções obtidas com a estratégia RCD - 40 clientes. .	139
Tabela 39 – Informações sobre as soluções obtidas com a estratégia CDR - 15 clientes. .	140
Tabela 40 – Informações sobre as soluções obtidas com a estratégia CDR - 25 clientes. .	141
Tabela 41 – Informações sobre as soluções obtidas com a estratégia CDR - 40 clientes. .	142
Tabela 42 – Informações sobre as soluções obtidas com a estratégia DD - 15 clientes. . .	143
Tabela 43 – Informações sobre as soluções obtidas com a estratégia DD - 25 clientes. . .	144
Tabela 44 – Informações sobre as soluções obtidas com a estratégia DD - 40 clientes. . .	145
Tabela 45 – Informações sobre as soluções obtidas com o modelo M2 - Cenário 1. . . .	150
Tabela 46 – Informações sobre as soluções obtidas com o modelo M2 - Cenário 2. . . .	151
Tabela 47 – Valor da função objetivo das soluções obtidas com as quatro versões da heurística construtiva - Cenário 1.	154
Tabela 48 – Tempo extra apresentado nas soluções obtidas com as quatro versões da heurística construtiva - Cenário 1.	155
Tabela 49 – Valor da função objetivo das soluções obtidas com as quatro versões da heurística construtiva - Cenário 2.	156
Tabela 50 – Valor da função objetivo e tempo de resolução das soluções obtidas com as duas versões da FO-VND - Cenário 1.	158
Tabela 51 – Valor da função objetivo e tempo de resolução das soluções obtidas com as duas versões da FO-VND - Cenário 2.	159

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Organização do texto	25
2	REVISÃO DA LITERATURA	27
2.1	Decisões operacionais	29
2.2	Características operacionais de <i>cross-docks</i>	29
2.2.1	<i>Cross-dock com uma doca</i>	31
2.2.2	<i>Cross-dock com múltiplas docas</i>	32
2.2.3	<i>Composição de cargas</i>	34
2.2.4	<i>Tipo de serviço do veículo</i>	34
2.3	Modelos da literatura	35
2.3.1	<i>Modelos baseados na ideia de precedência</i>	36
2.3.2	<i>Modelos baseados em designação</i>	39
2.4	<i>Cross-docking</i> e roteamento de veículos	41
2.4.1	<i>Problema de roteamento de veículos com cross-docking</i>	41
2.4.2	<i>Problema de roteamento de veículos com programação do cross-dock</i>	44
2.5	Considerações	45
3	PROBLEMA ESTUDADO: DEFINIÇÃO E MODELAGEM	49
3.1	Definição do problema	49
3.1.1	<i>Considerações</i>	51
3.2	Modelagem do problema	52
3.2.1	<i>Modelo M1</i>	54
3.2.2	<i>Modelo M2</i>	59
3.2.3	<i>Considerações</i>	60
3.2.3.1	<i>Consolidação das cargas e veículos de entrega</i>	61
3.2.3.2	<i>Função objetivo</i>	61
3.2.3.3	<i>Limite inferior para instante de término do carregamento</i>	61
3.3	Experimentos computacionais	62
3.3.1	<i>Geração das instâncias</i>	62
3.3.2	<i>Fase 1 - Análise dos modelos desenvolvidos</i>	65
3.3.3	<i>Fase 2 - Comparação com a literatura</i>	67
3.3.4	<i>Fase 3 - Relevância da integração dos problemas</i>	69

3.3.4.1	<i>Resultados para as instâncias com 15 clientes</i>	70
3.3.4.2	<i>Resultados para as instâncias com 25 clientes</i>	73
3.3.4.3	<i>Resultados para as instâncias com 40 clientes</i>	74
3.3.4.4	<i>Tempo de solução: $EI \times RCD$</i>	75
3.4	Conclusões	76
4	HEURÍSTICA CONSTRUTIVA	79
4.1	Algoritmo da heurística construtiva	79
4.2	Critério de Prioridade	80
4.3	Critério de Compatibilidade	81
4.4	Escolha da doca de saída	82
4.5	Sequenciamento de cargas	82
4.6	Inserção de cliente	83
4.7	Obtenção do valor da função objetivo	83
4.8	Experimentos computacionais	83
4.8.1	<i>Resultados computacionais - Cenário 1</i>	<i>84</i>
4.8.2	<i>Resultados computacionais - Cenário 2</i>	<i>85</i>
4.8.3	<i>Tempo de solução das instâncias</i>	<i>86</i>
4.9	Conclusões	87
5	META-HEURÍSTICA: BUSCA EM VIZINHANÇA VARIÁVEL	89
5.1	Algoritmo da meta-heurística	90
5.2	Modelo para as buscas locais	91
5.2.1	<i>Inequações válidas</i>	<i>92</i>
5.3	Vizinhanças utilizadas	93
5.3.1	<i>Vizinhança N_1 - designação e ordenação na saída</i>	<i>93</i>
5.3.2	<i>Vizinhança N_2 - designação e ordenação na entrada</i>	<i>94</i>
5.3.3	<i>Vizinhança N_3 - ordenação e roteamento em uma doca de saída</i>	<i>94</i>
5.3.4	<i>Vizinhança N_4 - designação, ordenação e roteamento em duas docas de saída</i>	<i>96</i>
5.3.5	<i>Vizinhança N_5 - Designação e ordenação de saída e roteamento para cargas em diferentes docas</i>	<i>97</i>
5.4	Experimentos computacionais	99
5.4.1	<i>Resultados computacionais - Fase 1</i>	<i>99</i>
5.4.2	<i>Resultados computacionais - Fase 2</i>	<i>104</i>
5.5	Conclusões	107
6	MATHEURÍSTICA: PROXIMITY SEARCH	109
6.1	Algoritmo da matheurística	110
6.2	Solução inicial	110

6.3	Modelo Auxiliar	112
6.4	Critérios de parada para resolução de MA	112
6.5	Experimentos computacionais	113
6.6	Conclusões	118
7	CONCLUSÕES E TRABALHOS FUTUROS	119
REFERÊNCIAS		123
APÊNDICE A	DETALHES SOBRE AS SOLUÇÕES OBTIDAS NO CAPÍTULO 3	129
A.1	Soluções obtidas com a relaxação linear dos dois modelos	129
A.2	Soluções obtidas com os dois modelos	132
A.3	Soluções obtidas com as estratégias propostas	134
A.3.1	<i>Estratégia integrada</i>	134
A.3.2	<i>Estratégia roteamento/cross-dock</i>	137
A.3.3	<i>Estratégia cross-dock/roteamento</i>	140
A.3.4	<i>Estratégia de entrega direta</i>	143
A.4	Gráfico de comparação entre as estratégias EI e RCD para as ins- tâncias com 15 clientes do Cenário 1.	145
APÊNDICE B	SOLUÇÕES PARA AVALIAÇÃO DAS SOLUÇÕES HEU- RÍSTICAS	149
APÊNDICE C	SOLUÇÕES OBTIDAS COM A HEURÍSTICA CONS- TRUTIVA	153
APÊNDICE D	SOLUÇÕES OBTIDAS COM A META-HEURÍSTICA	157

INTRODUÇÃO

Uma cadeia de suprimento (*supply chain*) é um sistema logístico complexo que permite transformar insumos em produtos finais (GHIANI; LAPORTE; MUSMANNO, 2004). Ela inclui fornecedores, fabricantes, clientes e centros de armazenamento e de distribuição, e suas principais atividades estão relacionadas à produção e à distribuição. De forma geral, a distribuição pode ser realizada por meio de entregas diretas ou passando por meio de instalações intermediárias, conforme ilustrado na Figura 1. No caso com instalações intermediárias, existem dois tipos de estratégias: *warehousing* e *cross-docking*. A diferença entre elas é que a estratégia *cross-docking* não permite estoque de produtos por longos períodos de tempo, enquanto que a *warehousing* permite.

Figura 1 – Ilustração das três estratégias de distribuição.

Fonte: Adaptada de Ghiani, Laporte e Musmanno (2004).

A estratégia *cross-docking* é apontada como uma alternativa para obter um sistema de distribuição com baixos custos de estoque e para melhor aproveitamento da capacidade de

transporte. Diversos exemplos de sua implementação em diferentes contextos são relatados na literatura. De acordo com [Mohtashami \(2015\)](#), o caso de estudo mais famoso é o da rede de varejo *Walmart*. [Boysen \(2010\)](#) cita exemplos no contexto de empresas postais, de fabricantes de automóveis e de transportadoras de cargas fracionadas. A estratégia consiste em transportar cargas de fornecedores para clientes passando por uma instalação intermediária onde as cargas são recebidas e separadas para consolidar novas cargas que são coletadas e entregues aos clientes. Além da redução de custos logísticos, essa estratégia contribui também para um sistema de distribuição dentro da filosofia *Just-in-Time* com entregas e tempos de respostas mais rápidos. Esta estratégia se encaixa no novo paradigma de cidade com base nos conceitos de *city logistic* que abordam questões ambientais e problemas de tráfego nas cidades, contribuindo para a redução de veículos e promovendo a mobilidade nos centros urbanos.

Os diversos trabalhos da literatura relacionados à estratégia *cross-docking* consideram decisões em níveis estratégico, tático ou operacional conforme pode ser verificado nos artigos de revisão de [Boysen e Fliedner \(2010\)](#) e de [Buijs, Vis e Carlo \(2014\)](#). Em especial, [Buijs, Vis e Carlo \(2014\)](#), analisando as variáveis de decisões consideradas nas diferentes abordagens, identificam 24 problemas relacionados ao *cross-docking* ou à rede da qual faz parte. Entre as direções de pesquisas apontadas pelos autores, ressaltam a importância da sincronização das decisões para as operações realizadas no *cross-dock* e na rede, em especial, apontam com as decisões para o roteamento de veículos.

Do ponto de vista operacional, a maioria dos trabalhos sobre *cross-docking* abordam o problema de programação de cargas no *cross-dock*, ou seja, as operações de descarregamento e de carregamento sob diversos aspectos. Por outro lado, a consolidação de cargas destinadas para diferentes clientes é uma estratégia logística bastante utilizada para o melhor aproveitamento da capacidade de transporte no qual é necessário tomar decisões sobre as rotas de forma a minimizar os custos com o transporte. Na prática, dependendo do sistema de distribuição, os problemas de programação de cargas no *cross-dock* e de roteamento podem ser relevantes para tomada de decisões no planejamento das atividades. Por exemplo, em um sistema de distribuição típico de uma rede de varejo que utiliza a estratégia *cross-docking*, os dois problemas podem ser tratados de forma integrada o que se configura em uma integração de decisões do *cross-docking* com decisões da rede. De acordo com [Yin, Lyu e Chuang \(2016\)](#), na prática, não faz sentido tratar esses dois problemas separadamente, é necessário abordá-los de forma integrada para garantir uma distribuição mais eficiente.

Neste trabalho, abordamos os problemas de programação de cargas no *cross-dock* (ou, simplesmente programação do *cross-dock*) e de roteamento de veículos baseado em um sistema de distribuição de uma rede varejo. No que é de conhecimento do autor, as abordagens da literatura que mais se aproximam deste trabalho são [Agustina, Lee e Piplani \(2014\)](#) e o trabalho recente de [Rahbari et al. \(2019\)](#). Nos dois trabalhos, os autores tratam problemas com características específicas da aplicação estudada, o que não permite aplicá-los diretamente ao contexto estudado

aqui. Desconsiderando as particularidades de cada trabalho, basicamente, a programação do *cross-dock* consiste em decidir a designação e a ordenação das cargas para as operações de descarregamento e de carregamento, porém, no problema estudado, as cargas que são carregadas e coletadas pelos veículos responsáveis pelas entregas podem conter pedidos para diferentes clientes localizados em pontos distintos e com janelas de tempo para as entregas e, por isso, é necessário também decidir as rotas de entrega. Assim, o objetivo do problema estudado é definir todas essas decisões de forma que o custo total para a distribuição seja mínimo.

Para o problema estudado, desenvolvemos um modelo integrando os problemas de programação do *cross-dock* e de roteamento de veículos. Com a adição de novas variáveis, um outro modelo alternativo com melhor relaxação linear é apresentado. Para mostrar a relevância da integração dos dois problemas, foi apresentada uma análise comparativa da estratégia integrada proposta com três estratégias comumente utilizadas na prática para resolução dos problemas. Em uma delas, admitimos apenas entregas diretas. Nas outras duas, resolvemos os dois problemas de forma hierárquica. Em uma, primeiro obtém-se a programação para *cross-dock* e, depois, as rotas são definidas. Na outra, determinamos as rotas primeiro para, então, definir a programação do *cross-dock*. De fato, os resultados obtidos com os experimentos computacionais mostram vantagens da estratégia integrada.

Como esperado, para instâncias grandes não é possível obter rapidamente soluções próximas ao ótimo para o problema integrado. Diante disso, propomos alguns métodos heurísticos para o problema estudado. Primeiro, desenvolvemos uma heurística construtiva com a qual são obtidas soluções em baixo tempo de processamento e, principalmente, para um dos cenários avaliados, soluções em média a 22,5% das soluções obtidas pelo *solver* de otimização Gurobi ao resolver o modelo com tempo de processamento bem maior. Além disso, propomos uma meta-heurística de busca em vizinhança variável com um bom desempenho, em particular, para as instâncias maiores tivemos ganhos de até 27,1% em um tempo computacional bem menor quando comparado ao Gurobi com o qual foram obtidas soluções em média a 23% do ótimo. Por último, apresentamos uma adaptação da matheurística *Proximity Search* para o problema que, em comparação ao *solver*, teve também um bom desempenho com ganhos de até 25% em relação a solução.

1.1 Organização do texto

Depois deste capítulo, o texto é organizado de acordo com a seguinte estrutura. No Capítulo 2, é apresentada uma revisão sobre o assunto estudado dividida em decisões operacionais, características operacionais de *cross-docks*, modelos para o problema de ordenação e/ou designação de cargas, trabalhos que abordam *cross-docking* e roteamento de veículos e considerações. No Capítulo 3, apresentamos com detalhes o problema estudado e os modelos desenvolvidos. Além disso, são reportados os resultados dos experimentos computacionais para a

comparação dos modelos e a análise da relevância da integração dos problemas de programação do *cross-dock* e de roteamento para o sistema de distribuição considerado. Nos capítulos seguintes, detalhamos os métodos heurísticos desenvolvidos. No Capítulo 4, descrevemos a heurística construtiva proposta. No Capítulo 5, propomos uma meta-heurística baseada em busca local em que usamos a solução obtida com a heurística construtiva como solução inicial. No Capítulo 6, desenvolvemos uma matheurística *Proximity Search* para tratar o problema. E, por último, as conclusões e trabalhos futuros são apresentados no Capítulo 7.

REVISÃO DA LITERATURA

A estratégia *cross-docking* tem sido apontada como uma das alternativas para obter um sistema de distribuição com baixos custos de estoque e um fluxo de produtos com entregas mais rápidas e, conseqüentemente, uma maior satisfação dos clientes. Diversos exemplos de implementação dessa estratégia em diferentes contextos são encontrados na literatura, sendo o caso de estudo mais famoso o da rede de varejo *Wal Mart* (VOGT, 2010; MOHTASHAMI, 2015). Em Boysen (2010), os autores também citam exemplos no contexto de empresas postais, de fabricantes de automóveis e de transportadoras de cargas fracionadas.

Figura 2 – Representação de um *cross-dock* e das atividades nele realizadas.

Fonte: Elaborada pelo autor.

Um sistema de distribuição *cross-docking* consiste em distribuir os produtos dos fornecedores para os clientes passando por uma instalação intermediária, denominada *cross-dock*, sem armazenamento de produtos ou, quando necessário, com armazenamento que, geralmente, não ultrapassa 24 horas. Nos *cross-docks*, as cargas enviadas pelos fornecedores são descarregadas nas docas de entrada e, imediatamente, separadas de acordo com os destinos finais para consolidar as novas cargas (cargas de saída) que são coletadas nas docas de saída e entregues

aos clientes. A Figura 2 ilustra um *cross-dock* com duas docas de entrada e três docas de saída. Note que, dentro da instalação, pode existir uma área de estoque temporário. Na figura, podemos visualizar três atividades que resumem as operações internas da instalação: descarregamento, separação (consolidação) e carregamento.

De acordo com Belle, Valckenaers e Cattrysse (2012), os centros de distribuição tradicionais (*warehousing*) apresentam, em geral, altos custos em virtude das instalações para o armazenamento e do processo de coleta dos produtos para consolidar cargas de saída. Diante disso, a maioria dos trabalhos na literatura destacam a redução ou a eliminação desses custos quando se utiliza a estratégia *cross-docking*. Outras vantagens como entregas em curtos intervalos de tempo, melhora na qualidade do serviço, aumento no volume de negociações e redução de avarias e perdas também são destacadas. Além disso, quando comparado à entrega direta, outras vantagens são destacadas, como, por exemplo, a redução de custos devido à consolidação de cargas, ao melhor aproveitamento dos recursos e do espaço (por exemplo, a capacidade de transportar) e a uma melhor correspondência entre as quantidades distribuídas e a demanda real, ou seja, um melhor atendimento dos clientes.

Segundo Agustina, Lee e Piplani (2010) e Buijs, Vis e Carlo (2014), as decisões associadas ao *cross-docking* podem ser divididas em estratégicas, táticas e operacionais. Para cada uma dessas categorias, as decisões podem estar direcionadas ao *cross-dock* ou à sua rede. As decisões estratégicas estão associadas à infraestrutura da rede ou do *cross-dock* e são tomadas a longo prazo, por exemplo, estabelecer a localização das instalações e definir a quantidade de docas suficiente para movimentação de um determinado fluxo de cargas. As decisões táticas referem-se ao planejamento de médio prazo como definir os recursos necessários e as configurações da instalação ou da rede para o funcionamento do sistema, bem como designar o tipo de serviço das docas ou estabelecer o fluxo das cargas de acordo com a capacidade de transporte da rede. E, por último, as decisões operacionais são tomadas no planejamento a curto prazo (diário) das atividades, por exemplo, designar e/ou ordenar cargas para carregamento e descarregamento e planejar o uso dos equipamentos dentro do *cross-dock* para separação e consolidação das cargas.

Dado que o foco deste trabalho são as decisões operacionais, a revisão da literatura aqui apresentada é direcionada a trabalhos que tratam questões operacionais. Informações sobre trabalhos que abordam, estrategicamente ou taticamente, o uso da estratégia *cross-docking* podem ser encontradas nos trabalhos de Agustina, Lee e Piplani (2010) e de Buijs, Vis e Carlo (2014). O primeiro trabalho contribui com uma revisão sobre os modelos matemáticos de acordo com o tipo de decisão e o segundo propõe uma classificação mais geral por incluir as decisões sincronizadas.

Esta revisão da literatura é distribuída da seguinte maneira. Primeiramente, apresentamos as decisões operacionais do *cross-dock* relacionadas ao problema estudado. Depois, descrevemos diferentes características operacionais de *cross-docks* que aparecem na literatura e são relevantes para as decisões operacionais. Na sequência, são identificados os trabalhos que propõem modelos

para a programação do *cross-dock* que mais se aproximam do trabalho desenvolvido aqui. Logo após, fazemos uma revisão sobre trabalhos que abordam o *cross-docking* e roteamento de veículos de forma integrada. E, por último, algumas considerações são apresentadas.

2.1 Decisões operacionais

A maioria dos trabalhos com foco operacional abordam decisões sobre como executar as operações para o descarregamento e para o carregamento de cargas. De acordo com [Boysen \(2010\)](#), decidir em que ordem e/ou designar as docas para o processamento (descarregamento ou carregamento) das cargas são questões que exigem respostas diárias nos *cross-docks* e influenciam diretamente na eficiência com que as operações são executadas no seu interior e, consequentemente, na eficiência do processo de distribuição para os clientes. Isto justifica o fato de que o principal assunto estudado no contexto operacional tem sido o problema de designação e/ou de ordenação de cargas. Os trabalhos da literatura abordam o problema sob diferentes aspectos determinados pelo tipo de *cross-dock* e suas características operacionais e pelo objetivo que determina a qualidade do planejamento ([BOYSEN, 2010](#)).

[Ladier e Alpan \(2016\)](#) ressaltam que o problema de planejar a ordem de processamento das cargas é referido apenas como problema de *scheduling* de cargas pela maioria dos autores e, quando o número de cargas é maior que a quantidade de docas, também está implícito o problema de designação que consiste em determinar em qual doca cada carga é processada. Os autores apresentam uma discussão sobre as diferenças entre os problemas de *sequencing* e de *scheduling*. O problema de *sequencing* consiste em determinar a ordem em que as cargas devem ser processadas e a própria ordenação define o instante de processamento, enquanto que no *scheduling* a ordem de processamento das cargas é determinada ao definir os instantes em que devem iniciar ou terminar os processamentos das cargas, logo o tempo é explicitamente determinado. Desta forma, usaremos a expressão ordenação para nos referirmos aos dois problemas sempre que não for necessária a distinção entre eles.

2.2 Características operacionais de *cross-docks*

Os *cross-docks* podem apresentar três formas de operação identificadas na literatura de acordo com [Belle, Valckenaers e Cattrysse \(2012\)](#). Na primeira, os produtos recebidos no *cross-dock*, imediatamente após serem descarregados nas docas de entrada, são enviados diretamente para a carga de saída posicionada em uma das docas para o carregamento. Nesta forma de operação, a transferência dos produtos é um processo, em geral, automatizado. A segunda forma é quando os produtos recebidos na instalação são enviados para uma área de consolidação próxima à doca de saída onde ficam temporariamente até serem carregados. E, por último, temos os *cross-docks* que recebem os produtos que também permanecem em uma área de estoque temporária, mas eles são reconfigurados para compor as cargas de saída, neste caso, existe

uma manipulação maior dos produtos dentro do *cross-dock*. Em especial, essa última forma de operação é a mais coerente no contexto de uma rede de varejo, na qual os pedidos são compostos por produtos de diferentes fornecedores (cargas de entrada), ou seja, uma rede de muitos para muitos.

Conforme já destacado, a ordenação e a designação de cargas são problemas frequentes para o planejamento das operações no *cross-dock*. De acordo com [Boysen \(2010\)](#), as decisões para estes problemas dependem das características operacionais do *cross-dock* que estão relacionadas ao tipo de operação da instalação, aos recursos disponíveis para as operações, à maneira como as operações podem ser executadas, às políticas adotadas e ao objetivo por meio do qual a qualidade do planejamento pode ser avaliada.

Tabela 1 – Resumo dos elementos e dos objetivos apresentados em [Ladier e Alpan \(2016\)](#).

Estratégico	Formato do <i>cross-dock</i> Número de docas Transporte interno
Tático	Modo de serviço da doca Preempção Capacidade de estoque temporário Capacidade dos recursos internos
Operacional	Tempo de chegada Tempo de partida Produtos intercambiáveis
Objetivo	Nível de estoque Horas de trabalho Balanço da carga de trabalho Tempo de viagem Congestionamento Tempo total de permanência dos produtos Tempo total de descarregamento ou carregamento <i>Makespan</i> Desvio de tempo para processamento Utilização das docas Produtos não carregados Custos de preempção Número de toques para manuseio dos produtos

Fonte: Elaborada pelo autor.

Em [Boysen \(2010\)](#), é apresentada uma classificação dos trabalhos que abordam o problema de ordenação e/ou de designação de cargas considerando o modo de serviço das docas, as características operacionais e os objetivos. [Belle, Valckenaers e Cattrysse \(2012\)](#) apresentam uma classificação mais geral com base em características físicas, operacionais e de fluxo de produtos. E o mais recente trabalho de revisão é de [Ladier e Alpan \(2016\)](#), no qual os autores fazem um levantamento dos elementos associados aos diferentes níveis de decisão (estratégico, tático e operacional). De acordo com os autores, esses elementos caracterizam um *cross-dock* e seus indicadores de performance (objetivos). Nesse trabalho, é apresentada uma comparação entre a literatura e a prática e os autores ressaltam que as decisões de longo e de médio prazo têm grande impacto sobre as operações nos *cross-docks*. Na Tabela 1, resumimos os elementos e os indicadores de performance listados em [Ladier e Alpan \(2016\)](#).

Dada que nossa revisão é restrita a trabalhos voltados ao planejamento operacional de um *cross-dock*, destacamos três elementos relevantes para nosso trabalho em que nos baseamos para a divisão das próximas subseções: número de docas, composição de produtos nas cargas (produtos intercambiáveis ou não) e tipo de serviço dos veículos. Em especial, dividimos o elemento número de docas em duas subseções, uma para os trabalhos que consideram *cross-dock* com uma doca e outra para os que consideram *cross-dock* com múltiplas docas.

2.2.1 *Cross-dock com uma doca*

Mesmo não sendo frequente na prática, muitos artigos abordam o problema com uma doca. Nestes casos, o problema de designação não aparece. [Ladier e Alpan \(2016\)](#) afirmam que estes estudos são importantes por darem direções de pesquisa para os casos mais complexos.

Em [Yu e Egbelu \(2008\)](#), o problema abordado consiste em encontrar a ordem para o carregamento e para o descarregamento das cargas considerando apenas uma doca de entrada e uma doca de saída de forma que o tempo total para executar todas as operações (*makespan*) seja mínimo. O *makespan* corresponde ao intervalo de tempo entre o instante em que a primeira carga começa a ser descarregada e o instante em que a última carga termina de ser carregada. Neste trabalho, admite-se uma área de estoque temporário ilimitada (suficiente para estocar os produtos temporariamente durante as operações). O mesmo problema é abordado em [Arabani, Ghomi e Zandieh \(2011\)](#), [Liao, Egbelu e Chang \(2012\)](#), [Shiguemoto, Netto e Bauab \(2014\)](#) e [Keshtzari, Naderi e Mehdizadeh \(2016\)](#) nos quais os autores desenvolveram métodos diferentes para resolvê-lo.

Problemas similares ao encontrado em [Yu e Egbelu \(2008\)](#) são tratados em outros trabalhos sob algumas características adicionais. [Forouharfard e Zandieh \(2010\)](#) determinam a ordenação das cargas tendo como objetivo minimizar o total de produtos que passam pela área de estoque temporário. [Soltani e Sadjadi \(2010\)](#) e [Vahdani, Soltani e Zandieh \(2010\)](#) não permitem o estoque temporário e admitem a preempção tanto no processo de descarregamento quanto no de carregamento, ou seja, uma carga que estiver sendo descarregada ou carregada pode ter seu processo interrompido para permitir o processamento de outra carga, mas a carga que teve seu processamento interrompido retorna a doca para ser concluído. O objetivo nestes dois trabalhos é a minimização de trocas que equivale a minimizar o *makespan*. [Arabani, Ghomi e Zandieh \(2010\)](#) introduzem os instantes em que as cargas de saída devem ser processadas e a sua ordenação é determinada de forma a minimizar a soma dos tempos de atraso e de antecipação. Além disso, abordam o problema como sendo multi-objetivo visando também a minimização do *makespan* e permitindo o estoque temporário.

Na Tabela 2, resumimos os objetivos e as características consideradas nas diferentes abordagens do problema apresentadas pelos trabalhos aqui citados que consideram apenas uma doca. Entre os objetivos, temos a minimização do *makespan* (MS), do estoque temporário (ET), das trocas (TR) quando a preempção é permitida e do tempo de antecipação e de atraso que

representa violação da janela de tempo (VJ). Sobre as características, identificamos a admissão do estoque temporário (ET) e da preempção (P).

Tabela 2 – Objetivos e características operacionais considerados em cada artigo revisado.

Artigo	Objetivo (minimização)				Características	
	MS	ET	TR	VJ	ET	P
Yu e Egbelu (2008)	✓				✓	
Arabani, Ghomi e Zandieh (2011)	✓				✓	
Liao, Egbelu e Chang (2012)	✓				✓	
Shiguemoto, Netto e Bauab (2014)	✓				✓	
Keshtzari, Naderi e Mehdizadeh (2016)	✓				✓	
Forouharfard e Zandieh (2010)		✓			✓	
Soltani e Sadjadi (2010)			✓			✓
Vahdani, Soltani e Zandieh (2010)		✓				✓
Arabani, Ghomi e Zandieh (2010)	✓			✓	✓	

MS: *makespan* ET: estoque temporário TR: trocas VJ: violação da janela de tempo P: preempção

Fonte: Elaborada pelo autor.

2.2.2 Cross-dock com múltiplas docas

No caso *cross-docks* com múltiplas docas para o processamento das cargas, alguns trabalhos estão focados em determinar a designação e a ordem para o processamento das cargas de entrada e das cargas de saída de forma simultânea. Com estas características, Li, Lim e Rodrigues (2004) e Chen e Song (2009) abordam o problema considerando uma área de estoque ilimitada e tendo como objetivo a minimização do *makespan*. Outros trabalhos, como Alvarez-Perez, González-Velarde e Fowler (2009) e Assadi e Bagheri (2016), consideram prazos para o processamento das cargas de entrada e de saída com o objetivo de minimizar a soma dos tempos de atraso e de antecipação que configuram as violações das janelas de tempo. Além de minimizarem essas violações, Ladier e Alpan (2018) minimizam também o número total de produtos que são enviados para a área de estoque temporária e Agustina, Lee e Piplani (2014) e Rahbari *et al.* (2019) minimizam os custos devido à permanência dos produtos dentro do *cross-dock*, em especial, Rahbari *et al.* (2019) tratam o problema bi-objetivo no qual também maximizam o frescor com que os produtos chegam aos seus destinos. Além disso, como Agustina, Lee e Piplani (2014) e Rahbari *et al.* (2019) integram a programação do *cross-dock* com o problema de roteamento de veículos no processo de distribuição das cargas de saída, os autores também consideram a minimização do custo de transporte. Já Boysen (2010) admite prazos apenas para as cargas de saída, minimizando a soma dos tempos de atraso sob um contexto de distribuição de alimentos congelados que não admite estoque temporário.

Em outros trabalhos, os autores preocupam-se apenas em determinar a ordenação das cargas de entrada, admitindo os prazos conhecidos para o despacho das cargas de saída. Entre

estes trabalhos estão Rosales, Fry e Radhakrishnan (2009), Boysen, Briskorn e Tschöke (2013), Liao, Egbelu e Chang (2013) e Belle *et al.* (2013) que consideram uma área de estoque ilimitada. Rosales, Fry e Radhakrishnan (2009) e Belle *et al.* (2013) visam o planejamento das operações de descarregamento que minimize as distâncias percorridas para a transferência dos produtos e, em especial, no segundo trabalho deseja-se um equilíbrio do tempo de trabalho entre as equipes que executam as operações. Em Boysen, Briskorn e Tschöke (2013) e em Liao, Egbelu e Chang (2013), o objetivo que mede a qualidade do planejamento é a minimização da quantidade de cargas de saída que não foram completamente carregadas, penalizadas no primeiro trabalho pela quantidade de produtos que não chega a tempo do carregamento e, no segundo, por um peso dependente do tipo de produto.

McWilliams, Stanfield e Geiger (2005), McWilliams, Stanfield e Geiger (2008), McWilliams (2010) e McWilliams e McBride (2012) também focam apenas no planejamento para o processamento de cargas de entrada, mas com um processo de transferência totalmente automatizado. Nestes trabalhos, o problema consiste em transferir os produtos por meio de esteiras que se conectam por alguns pontos, permitindo conduzir um produto de qualquer uma das docas de entrada para qualquer uma das docas de saída. O objetivo é determinar o planejamento de forma que o tempo de operação seja mínimo, o que, de acordo com os autores, é equivalente a minimizar o congestionamento dos produtos nos pontos de conexão durante o processo de transferência.

Na Tabela 3, resumimos os objetivos e as características consideradas pelos artigos citados nesta subseção. Além de alguns que aparecem nos trabalhos que consideram apenas uma doca, incluímos o número de cargas não completadas (CNC) e a distância percorrida (DP) para as transferências dos produtos nos objetivos e, nas características, as duas maneiras de tratar a ordenação das cargas, ordenação das cargas de saída e de entrada simultaneamente (*inbound and outbound scheduling* - IOS) ou apenas ordenação das cargas de entrada (*inbound scheduling* - IS). No caso dos trabalhos que integram a programação do *cross-dock* com o roteamento de veículos, temos também a minimização do custo de transporte (CT).

Tabela 3 – Objetivos e características operacionais considerados em cada artigo revisado.

Artigo	Objetivo (min.)						Características		
	MS	VJ	DP	CNC	ET	CT	ET	IOS	IS
Li, Lim e Rodrigues (2004)	✓			✓			✓	✓	
Chen e Song (2009)	✓			✓			✓	✓	
Alvarez-Perez, González-Velarde e Fowler (2009)		✓		✓			✓	✓	
Assadi e Bagheri (2016)		✓		✓			✓	✓	
Ladier e Alpan (2018)		✓			✓		✓	✓	
Agustina, Lee e Piplani (2014)		✓			✓	✓	✓	✓	
Rahbari <i>et al.</i> (2019)		✓			✓	✓	✓	✓	
Boysen (2010)	✓								✓
Rosales, Fry e Radhakrishnan (2009)			✓				✓		✓
Boysen, Briskorn e Tschöke (2013)				✓			✓		✓
Liao, Egbelu e Chang (2013)				✓			✓		✓
Belle <i>et al.</i> (2013)			✓				✓		✓
McWilliams, Stanfield e Geiger (2005)	✓								✓
McWilliams, Stanfield e Geiger (2008)	✓								✓
McWilliams (2010)	✓								✓
McWilliams e McBride (2012)	✓								✓

MS: *makespan* VJ: violação da janela de tempo DP: distância percorrida CNC: cargas não completadas
ET: estoque temporário IOS: ordenação na entrada e na saída IS: ordenação na entrada

Fonte: Elaborada pelo autor.

2.2.3 Composição de cargas

Em relação aos produtos descarregados, eles podem ser intercambiáveis ou não. De acordo com [Ladier e Alpan \(2016\)](#), dois produtos do mesmo tipo são intercambiáveis se um pode ser substituído pelo outro em uma determinada carga para satisfazer uma dada demanda. No caso de transferências de produtos intercambiáveis, os tipos de produtos e suas respectivas quantidades demandadas por uma carga de saída ou por um destino são conhecidos e as decisões sobre as transferências dentro da instalação são tomadas no descarregamento. Trabalhos como [Yu e Egbelu \(2008\)](#), [Joo e Kim \(2013\)](#) e [Ladier e Alpan \(2015\)](#) exemplificam este caso. Na transferência de produtos não intercambiáveis, tratado na maioria das abordagens, como, por exemplo, em [Chen e Song \(2009\)](#), em [Agustina, Lee e Piplani \(2014\)](#) e em [Rahbari *et al.* \(2019\)](#), cada produto tem um destino já definido e conhecido antes de ser descarregado.

2.2.4 Tipo de serviço do veículo

O tipo de serviço do veículo define a função atribuída a ele no *cross-dock* que pode ser entregar e/ou coletar. Esta característica não aparece no Tabela 1, mas consideramos importante ressaltá-la. Em geral, os trabalhos consideram que os veículos são destinados exclusivamente às entregas ou às coletas. Em especial, listamos dois trabalhos que consideram veículos mistos, ou seja, veículos que entregam e coletam cargas. Em [Joo e Kim \(2013\)](#), alguns veículos que entregam cargas também fazem coleta e, por isso, eles são considerados no planejamento das operações de descarregamento e de carregamento e, evidentemente, só podem ser carregados

depois que tiverem sido descarregados. [Shakeri et al. \(2012\)](#) abordam o problema de ordenação e de designação em um contexto em que as cargas são trocadas entre os veículos, isto é, os veículos que entregam cargas também coletam.

2.3 Modelos da literatura

Alguns trabalhos apresentam discussões ou provas sobre a complexidade do problema de ordenação de cargas considerando situações especiais e mais simples que os problemas encontrados na prática como pode ser verificado em [Chen e Lee \(2009\)](#), [Alvarez-Perez, González-Velarde e Fowler \(2009\)](#) e [Ladier e Alpan \(2014\)](#). Pode se observar que a complexidade aumenta à medida que a abordagem se aproxima de problemas reais. De acordo com [Ladier e Alpan \(2016\)](#), muitas abordagens propõem modelos bastante simplificados ficando distantes das necessidades reais.

Também em consequência da complexidade computacional, frequentemente são apresentadas heurísticas e meta-heurísticas para resolver os problemas. [Ladier e Alpan \(2016\)](#) classificam os trabalhos relacionados ao *cross-docking*, independente da abordagem ser estratégica, tática ou operacional, de acordo com o método de solução: exato, heurístico ou meta-heurístico. Considerando os métodos exatos de solução, em especial, programação matemática, o caminho mais natural é propor um modelo matemático para o problema. Muitas vezes, existem diferentes maneiras de formular um mesmo problema. Dependendo do método de solução exato, a qualidade da formulação pode melhorar seu desempenho na resolução do problema, como acontece nos métodos tipo *Branch and Cut*. Teoricamente, uma maneira de constatar se uma formulação é melhor que outra é mostrando qual se aproxima mais da envoltória convexa do problema estudado. Do ponto de vista computacional, outros fatores devem ser levados em consideração como, por exemplo, o número de variáveis inteiras e o número de restrições.

Para os problemas de designação e ordenação de cargas, diferentes modelos são propostos na literatura. Em geral, esse caminho não permite resolver instâncias de tamanho real, mas é importante para a avaliação de métodos heurísticos ou meta-heurísticos. Considerando as abordagens em que o problema é determinar o planejamento para o processamento das cargas de entrada e de saída de forma simultânea, podemos identificar três partes na estrutura dos modelos conforme o esquema ilustrado na Figura 3. Duas partes são compostas pelas restrições que modelam as decisões de designação e de ordenação no descarregamento e no carregamento, respectivamente, das cargas de entrada e de saída e a outra parte contém restrições que conectam essas decisões e, quando for o caso, restrições relacionadas às transferências de cargas.

Salvo as particularidades, os modelos para os problemas de designação e de ordenação de cargas apresentam semelhanças com os modelos para o problema de *scheduling* de tarefas em máquinas. De acordo com [Pan \(1997\)](#), três diferentes propostas de modelagem para o problema de *scheduling* de tarefas foram inicialmente apresentadas por [Wagner \(1959\)](#), [Bowman \(1959\)](#) e

Figura 3 – Representação da estrutura dos modelos que abordam o processamento de cargas de entrada e de saída de forma simultânea.

Fonte: Elaborada pelo autor.

[Manne \(1960\)](#). [Wagner \(1959\)](#) modela o problema de maneira que a ordem do processamento das tarefas em uma máquina é determinada pela designação destas tarefas para as posições de uma fila fictícia na máquina que podem representar intervalos de tempos diferentes de acordo com o tempo de processamento das tarefas a serem designadas. [Bowman \(1959\)](#) também modela o problema com base em designação de tarefas, mas, neste caso, a atribuição é feita para intervalos de tempo fixos, ou seja, o tempo é discretizado diferente de [Wagner \(1959\)](#) que considera o tempo contínuo. Por último, [Manne \(1960\)](#) propõe uma modelagem em que a ordem é estabelecida pela relação de precedência entre as tarefas.

Estas três estratégias de modelar o *scheduling* de tarefas em máquinas são identificadas nos modelos propostos para os problemas de ordenação e/ou de designação de cargas. A seguir, a revisão sobre os trabalhos que apresentam modelagens para estes problemas no planejamento simultâneo do processamento das cargas de entrada e de saída é dividida em duas subseções. Na primeira, são descritos trabalhos que apresentam modelos baseados na ideia de precedência, cujas estratégias para a modelagem são parecidas com a usada por [Manne \(1960\)](#) e, na segunda, são descritos trabalhos que apresentam modelos baseados em designação que usam estratégias de modelar similares às usadas por [Wagner \(1959\)](#) ou por [Bowman \(1959\)](#).

2.3.1 Modelos baseados na ideia de precedência

Nos modelos baseados na ideia de precedência, existem variáveis que indicam em qual doca cada carga deve ser processada e variáveis que determinam a relação de precedência entre as cargas processadas em uma mesma doca. Essa relação de precedência é estabelecida através de restrições disjuntivas modeladas com um *big-M* que, geralmente, piora a qualidade das relaxações lineares para a obtenção de limitantes duais em métodos do tipo *Branch and Cut*.

• Modelos de precedência para uma doca

O único modelo de precedência foi proposto por [Yu e Egbelu \(2008\)](#) para a ordenação de cargas com uma doca destinada exclusivamente ao descarregamento e outra para o carregamento.

O problema modelado consiste em obter uma sequência em que as cargas de entrada e de saída devem ser processadas tal que o *makespan* seja mínimo. Para a modelagem, os autores consideram o tempo de processamento proporcional à quantidade de produtos, o tempo de troca entre duas cargas consecutivas, o tempo de transferência da doca de entrada para a doca de saída e uma área de estoque temporário com capacidade ilimitada. Para a modelagem Problema de roteamento de veículos com programação do *cross-dock* - VRCDSP das decisões relacionadas à ordenação das cargas de entrada e de saída, são definidas restrições disjuntivas para todo par de cargas de entrada e de saída respectivamente, enquanto que, para a modelagem das restrições de conexão, são definidas restrições disjuntivas para cada par de carga considerando uma carga de entrada e outra carga de saída, além das restrições que determinam as transferências. Três tipos de variáveis binárias são definidos que, juntamente com as restrições disjuntivas, estabelecem a relação de precedência por meio dos instantes de tempo que são representados por variáveis contínuas. Os autores também propõem heurísticas para resolver o problema combinando três estratégias de seleção para a ordenação das cargas. Para avaliar as heurísticas são geradas instâncias randomicamente para diferentes quantidades de cargas de entrada e de saída e de tipos de produtos classificadas como pequenas e médias. A qualidade das soluções heurísticas são avaliadas através das soluções ótimas encontradas usando um método de enumeração completa.

Apesar de não terem outros modelos de precedência na literatura para o problema abordado por Yu e Egbelu (2008), existem trabalhos nos quais são propostos outros métodos para resolvê-lo. Vahdani e Zandieh (2010) desenvolveram cinco meta-heurísticas com as quais são realizados testes usando as mesmas instâncias apresentadas em Yu e Egbelu (2008) e outras 25 instâncias geradas pelos autores com quantidade maiores de cargas de entrada e de saída. Os resultados com a resolução destas instâncias mostram que as meta-heurísticas propostas em geral fornecem soluções com menor *makespan* apresentado pelas soluções encontradas pelas heurísticas propostas por Yu e Egbelu (2008). Em Arabani, Ghomi e Zandieh (2011) também são desenvolvidas cinco meta-heurísticas, cujos parâmetros são configurados pelo método de Taguchi, e testes são realizados com nove diferentes conjuntos de instâncias geradas randomicamente pelos autores. Arabani, Ghomi e Zandieh (2011) conseguem resultados melhores ao compararem com as heurísticas propostas em Yu e Egbelu (2008).

- **Modelos baseados na ideia de precedência para múltiplas docas**

Li, Lim e Rodrigues (2004) foram os primeiros a apresentarem um modelo baseado na ideia de precedência para o problema de ordenação e de designação de cargas com múltiplas docas. A abordagem é baseada no problema de programação de máquinas paralelas. O problema é determinar a sequência de processamento das cargas considerando prazos para o processamento de cada carga de forma a minimizar o tempo total de atraso e de antecipação no descarregamento e no carregamento. As restrições de conexão, diferentemente de Yu e Egbelu (2008), não incluem restrições disjuntivas pelo fato do problema admitir que as transferências das cargas de entrada

necessárias para consolidar as cargas de saída são conhecidas e, conseqüentemente, as relações de precedência entre cargas de entrada e cargas de saída também são conhecidas. Dois tipos de variáveis binárias com dois índices são definidos. Uma variável que indica a doca em que a carga deve ser processada, isto é, y_{ik} assume o valor um se a carga i é processada na doca k e zero para o caso contrário. E outra que indica a precedência imediata entre duas cargas, isto é, x_{ij} é igual a um se a carga i precede imediatamente a carga j e igual a zero para o caso contrário. No mesmo trabalho, é proposta uma reformulação do modelo redefinindo as variáveis binárias que indicam a precedência imediata como variáveis de três índices que indicam precedência não imediata, isto é, I_{ijk} é igual a um se a carga i precede (não imediatamente) a carga j na doca k e é igual a zero para o caso contrário. A variável y_{ik} permanece inalterada na reformulação. A reformulação proposta reduz a quantidade de restrições. Além disso, nesse mesmo trabalho também são propostas duas meta-heurísticas para resolver instâncias que foram geradas e cujas soluções são comparadas com as soluções obtidas pela resolução do modelo utilizando o *CPLEX*. Para o mesmo problema abordado em [Li, Lim e Rodrigues \(2004\)](#), [Alvarez-Perez, González-Velarde e Fowler \(2009\)](#) propõem uma meta-heurística híbrida combinando GRASP e Busca Tabu a partir da qual foram obtidos melhores resultados.

Em [Chen e Song \(2009\)](#), o problema abordado é interpretado como um problema de programação de máquinas paralelas em dois estágios, onde o primeiro estágio representa o descarregamento, o segundo representa o carregamento e as tarefas representam o processamento das cargas. O objetivo do problema é determinar a sequência para o processamento das tarefas (cargas) que minimize o *makespan*, respeitando as relações de precedência conhecidas entre as tarefas (cargas). Análogo ao modelo reformulado em [Li, Lim e Rodrigues \(2004\)](#), as restrições que determinam as transferências não são necessárias e as variáveis e as restrições disjuntivas são similares, mas os autores escrevem o modelo de forma compacta por usar o índice que indica o estágio e pelas restrições serem similares nos dois estágios. Também foram desenvolvidas heurísticas que foram avaliadas ao resolver instâncias pequenas e grandes. Para as instâncias pequenas, as soluções fornecidas pelas heurísticas são comparadas com as soluções obtidas ao resolver o modelo pelo *CPLEX* e, para as instâncias grandes, as soluções heurísticas são comparadas com um *lower bound*, cujo cálculo é proposto pelos autores.

[Belle et al. \(2013\)](#) e [Assadi e Bagheri \(2016\)](#) apresentam modelos similares sob determinados aspectos. Nestes dois trabalhos, são abordados problemas que consideram diferentes tempos para as transferências que dependem das docas de entrada e de saída. As decisões de designação e de ordenação são modeladas usando variáveis e restrições parecidas que formulam a relação de precedência (não imediata) e consideram janelas de tempo dentro das quais as cargas devem ser processadas. No entanto, os autores admitem que as janelas de tempo podem ser violadas gerando atraso ou antecipação. A diferença entre eles é que, no problema abordado por [Belle et al. \(2013\)](#), as transferências entre as cargas de entrada e de saída são conhecidas e o objetivo é minimizar a soma do tempo total para as transferências com o tempo total de antecipação e de atraso, enquanto que, no problema abordado por [Assadi e Bagheri \(2016\)](#), as

transferências fazem parte das decisões do problema e o objetivo é apenas minimizar o tempo total de antecipação e de atraso.

Assim como [Assadi e Bagheri \(2016\)](#), [Joo e Kim \(2013\)](#) consideram as decisões de transferências, mas sem contabilizar o tempo para o transporte interno entre as docas de entrada e de saída. O modelo é descrito com variáveis binárias análogas ao modelo reformulado apresentado em [Li, Lim e Rodrigues \(2004\)](#), mas, da maneira como as restrições são definidas, as variáveis indicam precedência imediata entre as cargas. O objetivo é a minimização do *makespan*. Os autores consideram que alguns veículos entregam e coletam cargas. Também apresentam duas meta-heurísticas (algoritmo evolutivo e algoritmo genético) que são avaliadas considerando dois operadores de mutação por meio de testes com instâncias geradas pelos autores e classificadas em pequenas e grandes. As soluções fornecidas pelas meta-heurísticas para as instâncias pequenas foram comparadas com as soluções ótimas encontradas ao resolver o modelo usando o CPLEX. Uma comparação entre os métodos propostos é apresentada ao resolver as instâncias grandes que estão divididas em quatro grupos de acordo com a quantidade total de cargas (cargas de entrada e cargas de saída).

2.3.2 Modelos baseados em designação

Os modelos baseados na ideia de designação podem ser divididos em dois tipos. O primeiro consiste em dividir o espaço de tempo contínuo em cada doca em posições para as quais as cargas podem ser designadas para serem processadas e o intervalo de tempo de uma posição corresponde ao tempo de processamento da carga que foi a ela designada. No segundo tipo de modelo, o espaço de tempo é discretizado em intervalos fixos para os quais as cargas podem ser designadas. Assim, os dois modelos definem a ordem de processamento das cargas a partir das designações e apresentam, respectivamente, modelagens com estratégias análogas às propostas por [Bowman \(1959\)](#) e por [Wagner \(1959\)](#) para o *scheduling* de tarefas em máquinas.

A vantagem dos modelos baseados na ideia de designação é que as restrições disjuntivas são eliminadas ou reduzidas. conforme verificados a seguir, os trabalhos que propõem os modelos são em menor número se comparado ao número de trabalhos que usam modelos na ideia de precedência.

- **Modelos baseados em designação para uma doca**

[Yu e Thapa \(2015\)](#) propõem um modelo de designação para um problema simplificado de ordenação de cargas em que o horizonte de planejamento é dividido em intervalos fixos de tempo, denominados *slots*. O intervalo de tempo correspondente a um *slot* é suficiente para processar qualquer carga, para realizar as trocas de cargas e para transferir os produtos. Assim, a ordenação do processamento das cargas é dada pelas designações sem a definição de restrições disjuntivas. Como são conhecidas todas as transferências necessárias entre as cargas de entrada e

de saída, as restrições que conectam a ordem no descarregamento com a ordem no carregamento impõem que toda carga de saída deve ser designada a uma posição que seja igual ou superior às posições das cargas de entrada das quais ela recebe produtos, não sendo necessárias restrições disjuntivas. O objetivo é minimizar o *makespan* em termos das diferenças absolutas entre os instantes no descarregamento e os instantes no carregamento. Além do modelo proposto, os autores apresentam um estudo teórico sobre a obtenção de um *lower bound* para o problema e uma estratégia de decomposição para sua solução.

Para o mesmo problema abordado em Yu e Egbelu (2008), Keshtzari, Naderi e Mehdizadeh (2016) apresentam um modelo com restrições de designação em que o horizonte de planejamento é dividido em posições para as docas de entrada e de saída, cujas quantidades são iguais à quantidade de cargas descarregadas e de cargas carregadas, respectivamente. Os intervalos de tempo destas posições são variáveis e dependem do tempo de processamento das cargas a elas atribuídas. Os autores reportam uma análise de complexidade do modelo proposto e verificam que a quantidade de restrições e de variáveis reais foi reduzida em relação ao modelo de precedência apresentado por Yu e Egbelu (2008), em especial, a quantidade de restrições disjuntivas. Os testes computacionais também comprovam uma maior eficiência do modelo na resolução das instâncias pequenas e médias usando o CPLEX. Também é proposta uma meta-heurística híbrida para o problema a partir da qual são obtidos resultados com melhor desempenho quando comparadas às meta-heurísticas apresentadas em Vahdani e Zandieh (2010) e Arabani, Ghomi e Zandieh (2011).

- **Modelo baseados em designação para múltiplas docas**

Os dois modelos baseados na ideia de designação propostos por Berghman *et al.* (2015) e Ladier e Alpan (2015) para o caso com duas docas consideram o horizonte de planejamento dividido em *slots* e não incluem decisões sobre a designação das cargas, apenas restringem que em um determinado *slot* o número de cargas a serem processadas deve ser menor ou igual à quantidade de docas disponíveis e não apresentam restrições disjuntivas.

Em Berghman *et al.* (2015), o problema consiste em determinar a ordem em que as cargas devem ser processadas de maneira a minimizar o tempo de utilização da área de estoque temporária, ou seja, minimiza a soma dos tempos que as transferências de uma carga de entrada para uma carga de saída permanece na área de estoque temporário, penalizadas pelas suas quantidades de paletes. Além de apresentar o modelo, os autores fazem um estudo comparativo entre dois cenários, um com docas exclusivas para o carregamento e para o descarregamento e outro com docas mistas que servem para as duas operações. Os autores concluem que a inclusão das docas mistas permite encontrar soluções viáveis para todas as instâncias do problema abordado utilizadas nos testes, enquanto que, com a utilização de apenas docas exclusivas, algumas instâncias são infactíveis.

Ladier e Alpan (2015) estudam o problema em que as cargas de entrada trazem pedidos organizados em paletes para diferentes clientes e os pedidos são acomodados nas cargas de saída de acordo com os clientes para os quais elas são destinadas. Neste trabalho, a ordem de processamento das cargas é determinada considerando características operacionais como a quantidade limitada de recursos (número de pessoas e equipamentos) para a movimentação dos produtos recebidos, imposição de janelas de tempo e de uma quantidade mínima que as cargas de saída devem receber. Além disso, o objetivo é minimizar a quantidade de produtos na área de estoque temporária e as antecipações e os atrasos no processamento das cargas são penalizados. Os autores também apresentam uma prova de que o problema é NP-difícil e um gerador de instâncias. Para resolução de instâncias grandes, são propostos métodos heurísticos baseados na resolução de modelos auxiliares e uma meta-heurística (Busca Tabu). Os testes mostram que os métodos heurísticos propostos são capazes de resolver instâncias grandes em baixo tempo computacional.

2.4 *Cross-docking e roteamento de veículos*

Na literatura, são encontrados dois tipos de abordagens que tratam problemas de roteamento de veículos e *cross-docking* de forma integrada. No primeiro, o problema é definido como o Problema de Roteamento com *Cross-Docking* (*Vehicle Routing Problem with Cross-Docking* – VRPCD). Este problema é semelhante ao problema de coletas e de entregas (*Pick up and Delivery Problem*), porém no VRPCD todas as coletas são realizadas primeiro e, depois, enviadas para o *cross-dock* onde são reorganizadas em cargas de saídas para serem entregues. O problema consiste em determinar as rotas para a realização das coletas e das entregas, sincronizando os instantes de chegada das cargas de entrada com os instantes de partida das cargas de saída. Em geral, estas abordagens não tratam de detalhes das operações internas do *cross-dock*. O segundo tipo de abordagem visa o planejamento das operações no *cross-dock* considerando a integração de decisões operacionais locais com as decisões do roteamento de veículos para as entregas. Este problema é denominado por Agustina, Lee e Piplani (2014) como Problema de Roteamento de Veículos e Programação de *Cross-Dock* (*Vehicle Routing and Cross-Dock Scheduling Problem* – VRCDS). A Figura 4 ilustra cada uma das abordagens. A esquerda da figura é ilustrado o VRPCD e à direita, o VRCDS.

A seguir, detalhamos os trabalhos que consideramos mais importantes relacionados ao VRPCD e ao VRCDS.

2.4.1 *Problema de roteamento de veículos com cross-docking*

Lee, Jung e Lee (2006) foram os primeiros a tratar o VRPCD. O problema descrito pelos autores consiste em determinar as rotas para as coletas e as rotas para as entregas dos pedidos iniciando e terminando em um *cross-dock*, sendo que cada rota deve ser concluída dentro de um

Figura 4 – Ilustrações das duas abordagens do problema integrado de roteamento e *cross-docking*.

Fonte: Adaptada de [Lee, Jung e Lee \(2006\)](#).

horizonte de planejamento. Além disso, o instante de chegada no *cross-dock* deve ser o mesmo para todas as rotas de coletas, assim as cargas coletadas estão simultaneamente disponíveis no *cross-dock* para iniciar o processo de consolidação das cargas de saída. Dessa forma, a programação das rotas de coletas e de entregas não levam em consideração as decisões locais no *cross-dock* relacionadas aos processos de descarregamento, de transferências e de carregamento, apenas as consolidações são definidas de acordo com as rotas. Outros trabalhos como [Liao, Lin e Shih \(2010\)](#) e [Vahdani et al. \(2012\)](#) propõem métodos heurísticos para resolução deste problema.

[Wen et al. \(2009\)](#) abordam o VRPCD no qual cada pedido tem janelas de tempo dentro das quais devem ser realizadas sua coleta e sua entrega. Nessa abordagem, as rotas com as coletas não precisam chegar no *cross-dock* simultaneamente. Os autores, por meio de variáveis que representam os instantes de término do descarregamento de cada carga de entrada, de disponibilidade dos pedidos para serem carregados e de início do carregamento de cada carga de saída, definem uma interdependência entre as cargas de entrada e as cargas de saída de forma que uma carga de saída só pode ser processada se todos os produtos que fazem parte da sua rota de entrega estiverem disponíveis. As decisões referentes à consolidação das cargas de saída que ocorre dentro do *cross-dock* são definidas de acordo com as decisões de roteamento dos veículos. Os autores propõem um algoritmo de busca tabu com o qual conseguem boas soluções (menos de 5% do ótimo) em baixo tempo de computacional. [Tarantilis \(2013\)](#) e [Morais, Mateus e Noronha \(2014\)](#) tratam sobre o mesmo problema abordado por [Wen et al. \(2009\)](#) propondo diferentes métodos de resolução. [Moghadam, Ghomi e Karimi \(2014\)](#) abordam também este problema, porém permitem a divisão de um pedido que, neste caso, pode ser entregue por mais de um veículo. [Santos, Mateus e Cunha \(2013\)](#) apresentam uma variante do VRPCD, incluindo na função objetivo o custo pela movimentação de um pedido dentro do *cross-dock* para descarregá-lo de um veículo e carregá-lo em outro, ou seja, quando um pedido não é coletado e entregue por um mesmo veículo. Esta variante também permite coletar e entregar um

pedido numa mesma rota sem passar pelo *cross-dock* e, por isso, é denominado por *Pickup and Delivery Problem with Cross-Docking* (PDPCD). Os autores apresentam um algoritmo *Branch and Price* por meio do qual são encontradas soluções a no máximo 3% do ótimo para instâncias geradas com base em dados reais. Mais recentemente, [Nikolopoulou et al. \(2017\)](#) apresentam um estudo comparativo em relação à eficiência das estratégias com e sem *cross-dock* e o impacto de diferentes características espaciais e temporais, como, por exemplo, distribuição geográfica, proximidade entre os pares de pontos de coleta e de entrega e restrições de capacidade, sobre as estratégias. Os autores desenvolvem um algoritmo de busca local competitivo comparado à literatura.

Alguns trabalhos na literatura de VRPCD, como [Dondo e Cerdá \(2013\)](#) e [Dondo e Cerdá \(2014\)](#), incorporam algumas características relacionadas às atividades de descarregamento e de carregamento que ocorrem no *cross-dock*. Primeiramente, em [Dondo e Cerdá \(2013\)](#), os autores consideram na modelagem os instantes em que as cargas são processadas sem restrições de número de docas e também a minimização do *makespan* além de custos de transporte. Em seguida, em [Dondo e Cerdá \(2014\)](#), os autores consideram um número de docas menor que o número de cargas a serem processadas e, conseqüentemente, as decisões de ordenação estão presentes. Esse mesmo problema também é abordado em [Dondo e Cerdá \(2015\)](#), porém os autores consideram veículos heterogêneos e tempos de deslocamento entre as docas de entrada e as docas de saída.

[Yin, Lyu e Chuang \(2016\)](#) e [Maknoon e Laporte \(2017\)](#) estudam o VRPCD com janelas de tempo, com características adicionais. No primeiro trabalho, os autores apresentam um modelo bi-objetivo e não linear para um problema no qual são consideradas decisões de designação e de ordenação das cargas e restrições relacionadas à emissão de CO_2 . Em [Maknoon e Laporte \(2017\)](#), o problema abordado apresenta uma rede com múltiplos *cross-docks* entre os quais são permitidas transferências para a consolidação de cargas de entregas.

Ainda neste contexto de coletas e de entregas utilizando *cross-docking*, [Dondo, Méndez e Cerdá \(2009\)](#) e [Dondo, Méndez e Cerdá \(2011\)](#) abordam um problema mais geral que denominam por problema de roteamento com *cross-dock* e gestão da cadeia de suprimento (*Vehicle Routing Problem with Cross-Dock in supply chain management* - VRPCD-SCM). Nesses dois trabalhos, diferente da maioria dos citados anteriormente em que se assume uma conexão única entre fornecedor e cliente (um para um), os autores consideram uma rede em que a relação entre fornecedores e clientes é de muitos para muitos e as estratégias de distribuição são mistas (entrega direta, via *cross-docks*). Além disso, a rede também conta com pontos apenas para coletas ou apenas para entregas e outros para coletas e entregas. Os autores citam exemplos de problemas reais que se enquadram nessa abordagem, como uma cadeia de suprimentos de produtos refinados do petróleo ou de produtos químicos como, por exemplo, agrotóxicos. Em geral, os problemas não têm um número grande de produtos, como pode ser identificado nas instâncias reais resolvidas com quatro tipos de produtos.

2.4.2 *Problema de roteamento de veículos com programação do cross-dock*

Agustina, Lee e Iplani (2014) apresentam a primeira abordagem do VRCDSP. Os autores consideram um sistema de distribuição de alimentos perecíveis. Nesse trabalho, a definição das rotas de entrega aos clientes e a programação do *cross-dock* são consideradas de forma integrada. O problema consiste em um conjunto de clientes que recebem seus pedidos de apenas um fornecedor, mas cada fornecedor pode servir muitos clientes configurando, portanto, uma rede de distribuição de um para muitos. Além disso, os fornecedores organizam os pedidos em paletes e os enviam para o *cross-dock*, onde eles são separados e consolidados em cargas de saída. Cada carga de saída tem pedidos de diferentes clientes e, por isso, uma rota é definida para que sejam realizadas todas as entregas tentando respeitar as janelas de tempo. Assim, é preciso decidir sobre a designação e a ordenação para o descarregamento e o carregamento das cargas e sobre as rotas para distribuição. O objetivo é minimizar o custo total que é a soma dos custos de estoque temporário dos produtos e dos custos de transporte e custos pelas violações (antecipação ou atraso) das janelas de tempo para de entrega aos clientes. Os autores propõem um modelo de programação inteira mista para tratar todas as decisões de forma integrada com o qual são resolvidas instâncias pequenas com até 50 clientes, 4 fornecedores e 4 docas. Para as instâncias grandes com até 300 clientes, 20 fornecedores e 20 docas, uma outra formulação que considera o agrupamento de clientes por zonas (*clusters*) e janelas de tempos rígidas é usada. Nos dois casos, foi utilizado o *solver* CPLEX 12.3 para a resolução das instâncias e os tempos máximos reportados pelos autores são iguais a 1096,8 e 4026,8 segundos, respectivamente.

Recentemente, Rahbari *et al.* (2019) estendem o problema estudado por Agustina, Lee e Iplani (2014) considerando uma rede de distribuição de muitos para muitos e uma frota de veículos heterogêneos. Além disso, os autores apresentam uma abordagem bi-objetiva em que, além da minimização dos custos de estoque, de transporte e de violação das janelas de tempo nas entregas, deseja-se aumentar a rapidez com que os produtos são entregues aos clientes que é incorporado no modelo pela maximização do frescor com que o produto chega ao cliente. Esse frescor é medido por um índice que decresce com o passar do tempo após o produto ter sido descarregado no *cross-dock*. Os autores destacam que, no contexto de distribuição de produtos perecíveis, é muito importante que os produtos cheguem aos seus destinos em boa qualidade. Além do modelo desenvolvido para o problema com base no modelo apresentado por Agustina, Lee e Iplani (2014), os autores desenvolvem outros dois com incertezas. Em um deles, considera-se incertezas nos tempos de viagem e no outro as incertezas são sobre o tempo de validade dos produtos. Os testes realizados consideram instâncias pequenas e são voltados apenas para as análises sobre a importância de considerar a maximização do frescor dos produtos entregues e do impacto das incertezas sobre os custos.

2.5 Considerações

De acordo com a revisão, existem diversas abordagens que envolvem a estratégia de distribuição *cross-docking* com diferentes características e objetivos dependendo do contexto de aplicação. Mesmo assim, como [Ladier e Alpan \(2016\)](#) afirmam, são necessárias abordagens que se aproximam mais da prática. Em especial, [Buijs, Vis e Carlo \(2014\)](#) apontaram a importância de sincronizar decisões do *cross-dock* e da rede para obter uma sistema de distribuição mais eficiente. De acordo com [Yin, Lyu e Chuang \(2016\)](#), na prática, não faz sentido considerar os problemas de roteamento de veículos e de programação do *cross-dock* separadamente quando eles estão presentes em um sistema de distribuição. De certa forma, a integração entre esses problemas pode viabilizar consolidações de carga que permitem um melhor aproveitamento da capacidade de transporte e dá mais flexibilidade ao sistema. Nesse sentido, identificamos que começam a surgir trabalhos na literatura que abordam o VRCDSP.

O problema estudado neste trabalho, descrito no próximo capítulo, é o VRCDSP baseado em um sistema de distribuição de uma rede de varejo, no qual não são distribuídos produtos perecíveis. Nesse contexto, temos muitos fornecedores que enviam diferentes tipos de produtos para compor muitos pedidos que deverão ser entregues, configurando, portanto, uma rede de distribuição de muitos para muitos. Os pedidos são organizados em paletes dentro do *cross-dock* que irão consolidar as cargas de saída. Como cada carga de saída tem pedidos de diferentes clientes, é necessário definir as rotas para as entregas dos pedidos. Conforme apontado na revisão da literatura, no que diz respeito à integração dos problemas de roteamento de veículo e de programação do *cross-dock*, os dois trabalhos que mais se aproximam do nosso são [Agustina, Lee e Piplani \(2014\)](#) e [Rahbari et al. \(2019\)](#), porém eles são desenvolvidos com base em um contexto diferente do nosso que é a distribuição de produtos perecíveis e, consequentemente, apresentam diferenças em relação aos aspectos operacionais. A seguir, na Tabela 4, resumimos as diferenças mais relevantes entre estes trabalhos e o trabalho desenvolvido.

Tabela 4 – Diferenças entre a literatura e o problema estudado.

Aspecto	Agustina, Lee e Piplani (2014)	Rahbari et al. (2019)	Este trabalho
Tipo de rede	um→muitos	muitos→muitos	muitos→muitos
Tempo de processamento	igual	distintos	distintos
Tempo de transferência interna	não	não	sim e distintos
Número de veículos	conhecido	conhecido	decisão
Tempo de estoque	por pedido	por pedido	<i>makespan</i>

Fonte: Elaborada pelo autor.

As diferenças entre os problemas podem refletir diretamente na sua complexidade e formulação. No caso do VRCDSP, assim como é considerado em [Rahbari et al. \(2019\)](#), temos muitos pedidos que são compostos por produtos envidados por muitos fornecedores (muitos→muitos) e, por isso, eles são organizados dentro do *cross-dock* para então consolidar as cargas de saída. Além disso, igualmente em [Rahbari et al. \(2019\)](#), consideramos tempos de

processamento diferentes tanto no descarregamento quanto no carregamento, pois algumas cargas apresentam volumes maiores ou exigem mais trabalho. No nosso caso, por alguns produtos serem mais pesados ou por requererem mais cuidados, assumimos tempos para as transferências interna distintos de acordo com o tipo de produto e com as posições de descarregamento e de carregamento. Já [Agustina, Lee e Piplani \(2014\)](#) e [Rahbari et al. \(2019\)](#) não consideram esse tempo. Outra diferença é que o número de veículos é uma decisão importante para o problema estudado, enquanto que, para os outros dois trabalhos, esse número é conhecido. Como os produtos distribuídos não são perecíveis em nosso problema, eles podem esperar pelo seu carregamento sem prejuízos, consequentemente, não temos custos de estoque temporário. Assim, consideramos a minimização do *makespan* que, de acordo com [Belle, Valckenaers e Cattrysse \(2012\)](#), pode representar a mão-de-obra exigida. No caso da distribuição de produtos perecíveis, existem custos para manter o estoque temporário e por isso são mais relevantes.

Em relação à modelagem, identificamos três estratégias para modelar o problema de ordenação e/ou designação de cargas no *cross-dock* que são semelhantes às estratégias de modelagem para o problema de *scheduling* de tarefas em máquinas propostas, cada uma, por [Wagner \(1959\)](#), [Bowman \(1959\)](#) e [Manne \(1960\)](#). As estratégias propostas por [Wagner \(1959\)](#) e por [Bowman \(1959\)](#) modelam o problema de *scheduling* de tarefas por meio de variáveis e de restrições de designação considerando, respectivamente, o tempo contínuo e o tempo discretizado. Enquanto a estratégia proposta por [Manne \(1960\)](#) modela o problema por meio da relação de precedência entre as tarefas. Na Tabela 5 organizamos os trabalhos revisados que abordam o problema de ordenação e/ou designação de cargas de entrada e de saída de acordo com a estratégia de modelagem utilizada e separados pela quantidade de docas.

Tabela 5 – Classificação dos trabalhos de acordo com a estratégia de modelagem separados pela quantidade de docas.

Estratégia	Uma doca	Múltiplas docas
Manne	Yu e Egbelu (2008)	Li, Lim e Rodrigues (2004)
		Chen e Song (2009)
		Belle et al. (2013)
		Joo e Kim (2013)
		Agustina, Lee e Piplani (2014)
		Assadi e Bagheri (2016)
Bowman	Yu e Thapa (2015)	Rahbari et al. (2019)
Bowman	Yu e Thapa (2015)	Berghman et al. (2015)
		Ladier e Alpan (2016)
Wagner	Keshtzari, Naderi e Mehdizadeh (2016)	—

Fonte: Elaborada pelo autor.

Em especial, [Keshtzari, Naderi e Mehdizadeh \(2016\)](#) mostrou que o modelo baseado em designação é mais eficiente que o modelo baseado na relação de precedência para o caso com uma doca proposto por [Yu e Egbelu \(2008\)](#). Os poucos modelos com restrições de designação

para caso de múltiplas docas são propostos sob a simplificação de que toda carga de entrada e de saída tem um mesmo tempo de processamento. Esta simplificação não é válida para todas as situações. Por exemplo, cargas com maior quantidade ou complexidade de manuseio podem levar mais tempo para serem processadas que outras.

Na primeira parte deste trabalho, desenvolvemos dois modelos de programação inteira mista para o problema estudado. Um modelo é baseado na ideia de designação e o outro na ideia de precedência. Testes computacionais preliminares foram realizados e apontaram um melhor desempenho do modelo baseado na ideia de precedência que está disponível em [Bernardes, Armentano e Toledo \(2017\)](#)¹. Nos dois modelos inicialmente desenvolvidos, as decisões de roteamento foram modeladas com variáveis de três índices e não eram considerados os tempos de processamentos distintos, os tempos para as transferências interna de acordo com o tipo de produto e a posições de descarregamento e de carregamento e o *makespan*. Porém, esses pontos são importantes para o problema estudado. Assim, para incluí-los, um novo modelo foi desenvolvido com base na ideia de precedência e com a formulação de dois índices para a parte de roteamento que é apresentando no próximo capítulo, onde também definimos com detalhes o problema e apresentamos os resultados dos experimentos computacionais.

¹ Trabalho completo publicado nos Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional. Disponível em: <http://www.sbp2017.iltc.br/pdf/168858.pdf>.

PROBLEMA ESTUDADO: DEFINIÇÃO E MODELAGEM

O problema estudado aborda o planejamento operacional de um sistema de distribuição que utiliza a estratégia *cross-docking* integrando decisões de designação, de ordenação e de consolidação de cargas. Para o caso estudado, também são relevantes as decisões de roteamento de veículos no processo de entrega das cargas. Embora relevante na prática, conforme apontado na revisão bibliográfica, encontramos poucos trabalhos que tratam o problema de programação do *cross-dock* com roteamento de veículos. Os trabalhos que mais se assemelham ao problema estudado são: [Agustina, Lee e Piplani \(2014\)](#) e [Rahbari *et al.* \(2019\)](#). Os autores abordam um sistema de distribuição de produtos perecíveis com características distintas do contexto no qual nos baseamos para este trabalho. Neste capítulo¹, apresentamos uma definição detalhada do problema estudado e propomos dois modelos de programação inteira mista para representá-lo. Na sequência, reportamos os resultados de testes computacionais realizados para avaliar os modelos desenvolvidos e para analisar a relevância de tratar de forma integrada os problemas de programação do *cross-dock* e roteamento de veículos para a entrega das cargas.

3.1 Definição do problema

O sistema de distribuição considerado é composto por um *cross-dock* que recebe cargas com um ou mais tipos de produtos que são separados para compor os pedidos de um conjunto de clientes. Na Figura 5, temos um exemplo de um conjunto de cargas de entrada enviadas pelos fornecedores e um conjunto de clientes e seus respectivos pedidos. Note que cada pedido pode demandar produtos de muitas cargas de entrada e cada carga de entrada serve muitos clientes o que define uma rede de distribuição de muitos para muitos (muitos→muitos).

¹ O estudo apresentado neste capítulo foi compilado em um artigo que está em revisão pelo periódico *International Journal of Production Economics*.

Figura 5 – Conjuntos de cargas de entrada e de clientes e seus pedidos.

Fonte: Elaborada pelo autor.

As cargas recebidas são descarregadas nas docas de entrada e seus produtos são separados e organizados em paletes de acordo com os pedidos dos clientes para consolidar as cargas de saída, que serão coletadas pelos veículos de entrega nas docas de saída. O número de paletes necessário para organizar a demanda de cada cliente é conhecido a priori e é sempre menor ou igual a capacidade dos veículos responsáveis pelas entregas que podem ser realizada diretamente para o cliente ou por meio de rotas. Esse número é definido com base no volume total de produtos demandados e no volume que cada paleta suporta. As decisões do problema consistem em programar as atividades de descarregamento, de consolidação e de carregamento das cargas. Além disso, devem ser definidas as rotas para as entregas considerando o tempo de transferência dos produtos dentro do *cross-dock* e as janelas de tempo de cada cliente. Os tempos de transferência interna dependem da complexidade de manuseio dos diferentes tipos de produtos e das posições (docas) em que são descarregados e carregados.

Na Figura 6, ilustramos um exemplo do sistema de distribuição estudado. Da esquerda para a direita, representamos o conjunto de cargas recebidas dos fornecedores (cargas de entrada), o *cross-dock*, o conjunto de veículos homogêneos usados para as entregas e o conjunto de clientes, nessa ordem. O número de cargas recebidas e o número de cargas de saída consolidadas são maiores que o número de docas de entrada e de saída, respectivamente. As distâncias entre as docas de entrada e as docas de saída são diferentes. Além disso, alguns produtos têm tempos de transferência maiores (por exemplo, produtos mais pesados ou que requerem maior cuidado durante a transferência exigem maior tempo). As linhas que ligam os veículos e os clientes exemplificam as rotas de entregas. Dessa forma, as decisões a serem tomadas para a programação do sistema de distribuição estão relacionadas à designação e à ordenação das cargas para as operações de descarregamento e de carregamento sem preempção (i.e., quando o processamento de uma carga inicia, ele não pode ser interrompido). Destacamos que os tempos de processamento

Figura 6 – Representação de uma estrutura interna do *cross-dock* com uma ordenação e consolidação de cargas definidas.

Fonte: Elaborada pelo autor.

das cargas são também distintos. Além disso, é necessário definir as cargas de saída consolidadas com base nos clientes que fazem parte de uma mesma rota de entrega que também é uma decisão a ser tomada. Note que o número de cargas de saída define o número de veículos de entrega.

O objetivo do problema é obter uma programação para o *cross-dock* e um plano de distribuição que minimiza o custo total. O custo total inclui o custo operacional do *cross-dock* que está associado ao *makespan*, o custo de transporte e o custo associado ao atraso ou a antecipação da entrega dos pedidos aos clientes.

3.1.1 Considerações

Quando analisamos o sistema de distribuição como um todo, verificamos que existe uma relação de interdependência entre as atividades, como ilustrado na Figura 7. Os instantes das entregas dependem de como as operações são realizadas no *cross-dock*, especialmente de quando as cargas de saída estão disponíveis. As decisões de designação e de ordenação para o descarregamento das cargas de entrada refletem diretamente na disponibilidade das cargas de saída. Da mesma forma, a disponibilidade das cargas de saída reflete no instante de partida dos veículos que irão coletá-las. Por sua vez, os instantes de partidas desses veículos são importantes na definição das rotas para as entregas, consequentemente, nos instantes em que as entregas ocorrem. Por outro lado, quando as rotas são estabelecidas, elas determinam a consolidação das cargas de saída e a ordem do carregamento e, consequentemente, a ordem das cargas de entrada no descarregamento. Diante disso, a melhor estratégia é tomar as decisões de forma integrada.

Assim, conceitualmente, o problema descrito neste trabalho consiste na integração de dois problemas clássicos da literatura: a ordenação e designação de cargas no *cross-dock* com

Figura 7 – Representação da interdependência entre as decisões para obter um plano de distribuição.

Fonte: Elaborada pelo autor.

o roteamento de veículos com janela de tempo. Essa abordagem visa um planejamento que viabilize um número maior de entregas dentro dos prazos. Além disso, contribui para um sistema de distribuição mais dinâmico em relação à consolidação de cargas e a determinação de rotas de acordo com as demandas dos clientes que variam a cada horizonte de planejamento em que a distribuição é realizada, por exemplo, na tomada de decisões diária. Nesse sentido, propomos o modelo matemático apresentado a seguir e realizamos testes computacionais para evidenciar a relevância de tratar o problema integrado de roteamento de veículos e programação do *cross-dock* (do inglês *vehicle routing and cross-dock scheduling problem* - VRCDSP).

3.2 Modelagem do problema

Nesta seção, apresentamos dois modelos de programação inteira mista para o VRCDSP estudado. A diferença entre eles é a inclusão de uma variável que indica a rota a que cada cliente pertence e, conseqüentemente, algumas modificações nas restrições. Embora a nossa abordagem do VRCDSP seja baseada em um contexto de aplicação diferente do encontrado na literatura, os modelos propostos têm estruturas semelhantes às estruturas dos modelos apresentados em [Agustina, Lee e Piplani \(2014\)](#) e em [Rahbari *et al.* \(2019\)](#).

Figura 8 – Representação da estrutura dos modelos desenvolvidos para o VRCDSP.

Fonte: Elaborada pelo autor.

Na Figura 8, ilustramos a estrutura dos modelos que pode ser dividida em quatro sub-

conjuntos de restrições de acordo com os tipos de decisões. Dois conjuntos de restrições estão relacionados à designação e à ordenação para os processamentos das cargas de entrada e de saída, outro refere-se ao conjunto de restrições relativas ao roteamento e o quarto é o conjunto de restrições que estabelece as conexões entre os três conjuntos anteriores, definindo as transferências, as consolidações de cargas e relacionando os tempos de descarregamento, de carregamento e de entrega.

Primeiramente, listamos as hipóteses admitidas para a modelagem do problema.

- Todas as cargas de entrada estão disponíveis para a programação no início das operações.
- Os tempos de processamento para as cargas de entrada podem ser distintos e incluem os tempos para posicionar a carga na doca, para descarregar os paletes, para desmontá-los, quando necessário, e para identificar os produtos.
- O tempo para transferir uma unidade de cada tipo de produto é conhecido. Os tempos podem ser distintos dependendo da complexidade de manuseio e das posições de descarregamento e de carregamento de cada produto.
- O tempo de processamento de uma carga de saída corresponde ao tempo necessário para transportar os paletes consolidados da área de estoque temporário para o interior do veículo. Esse tempo é proporcional ao número de paletes da carga consolidada. Além disso, o tempo para a troca de veículos em uma determinada doca de saída é conhecido e independe da sequência dos carregamentos.
- Se o processamento de uma carga de entrada ou de saída for iniciado, ele não pode ser interrompido, ou seja, ele deve ser executado continuamente até o final (não é permitida preempção).
- Todos os produtos de uma carga de entrada estão disponíveis para as transferências interna somente depois que o processamento dessa carga for concluído,
- O número de paletes necessário para organizar a demanda de cada cliente é conhecido e sempre menor ou igual a capacidade do veículo de entrega.
- A frota de veículos utilizada para as entregas é homogênea, ou seja, todos os veículos têm a mesma capacidade em paletes e apresentam a mesma velocidade de deslocamento.
- Não é permitido o *split* de demandas. Assim, cada cliente recebe a entrega somente por um veículo.
- O custo da viagem entre dois clientes é diretamente proporcional ao tempo de viagem.
- O tempo de serviço em cada cliente para descarregar a entrega é proporcional ao número de paletes do seu pedido.

- As janelas de tempo para as entregas aos clientes são conhecidas, mas são permitidos atrasos e antecipações que são penalizados por uma unidade de custo para cada unidade de tempo atrasada ou antecipada nas entregas.
- Todas as operações para realizar a distribuição devem ser executadas dentro de um intervalo de tempo pré-determinado.

3.2.1 Modelo M1

Primeiramente, reportamos os conjuntos, os índices, os dados e as variáveis utilizados na primeira modelagem do problema, chamada de M1.

Conjuntos:

- $\mathcal{L}$: conjunto de L cargas de entrada,
 $\mathcal{P}$: conjunto de P tipos de produtos,
 $\mathcal{N}$: conjunto de N clientes,
 $\mathcal{F}$: conjunto de F docas de entrada,
 $\mathcal{H}$: conjunto de H docas de saída.

Índices:

- l, m : cargas de entrada,
 p : tipo de produto,
 i, j, n : cliente,
 f : doca de entrada,
 h : doca de saída.

Dados:

- PT_l : tempo para o processamento da carga de entrada l ,
 R_l^p : número de produtos do tipo p na carga de entrada l ,
 TR_{fh}^p : tempo necessário para a transferência de uma unidade do produto p da doca de entrada f para a doca de saída h ,
 LT : tempo para mover um palete da área de consolidação para o interior do veículo,
 CT : tempo de troca dos veículos na doca de saída,
 CAP : capacidade (em paletes) de cada veículo de entrega,
 D_i^p : número de produtos p demandados pelo cliente i ,
 Q_i : número de paletes necessário para organizar o pedido do cliente i ,
 CO : custo operacional do *cross-dock* por unidade de tempo,
 $TT_{ij} (CT_{ij})$: tempo (custo) de viagem entre o cliente i e o cliente j (ou entre um cliente e o *cross-dock* quando i ou j assumir 0 ou $N + 1$),

- CE (CL): penalidade por unidade de tempo de antecipação (atraso) nas entregas,
 ST_i : tempo de serviço para descarregar os paletes do pedido do cliente i ,
 $[A_i, B_i]$: janela de tempo para entrega do pedido do cliente i , em que A_i é o início da janela e B_i é o final,
 T^{max} : tempo máximo para completar todas as atividades do *cross-dock*, inclusive as entregas das cargas e o retorno dos veículos ao *cross-dock*,
 M : é um número suficientemente grande (assumimos $M = T^{max}$).

Variáveis binárias:

- $z_{lf}^{in} \in \{0, 1\}$: assume o valor 1, se a carga de entrada l é processada na doca de entrada f , e 0, caso contrário,
 $w_{lm}^{in} \in \{0, 1\}$: assume o valor 1, se as cargas de entrada l e m são processadas na mesma doca de entrada e o processamento da l precede o processamento da carga m , e 0, caso contrário,
 $z_{ih}^{out} \in \{0, 1\}$: assume o valor 1, se o veículo, cujo primeiro cliente visitado é o cliente i , coleta sua carga na doca de saída h , e 0, caso contrário,
 $w_{ij}^{out} \in \{0, 1\}$: assume o valor 1, se os veículos, que iniciam suas rotas nos clientes i e j , coletam suas cargas na mesma doca, sendo que o veículo que inicia a rota em i coleta antes que o veículo que inicia a rota em j , e 0, caso contrário,
 $y_{li} \in \{0, 1\}$: assume o valor 1, se pelo menos um tipo de produto é transferido da carga de entrada l para a carga de saída coletada pelo veículo que inicia sua rota em i , e 0, caso contrário,
 $x_{ij} \in \{0, 1\}$: assume o valor 1, se o veículo vai do cliente i para o cliente j , e 0, caso contrário.

Variáveis contínuas:

- $ut_l \geq 0$: instante de encerramento do processamento da carga de entrada l ,
 $rt_i \geq 0$: instante de liberação para o carregamento da carga de saída coletada pelo veículo cuja rota inicia com a entrega do cliente i ,
 $dt_i \geq 0$: instante de encerramento da operação de carregamento do veículo cuja rota inicia com a entrega do cliente i ,
 $dt^{max} \geq 0$: instante de encerramento da operação de carregamento do último veículo de entrega (equivalente ao *makespan* do *cross-dock*),
 $\rho_{lfih}^p \geq 0$: número de produtos p transferidos da carga de entrada l recebida na doca de entrada f para a carga de saída coletada pelo veículo cuja rota inicia com a entrega do cliente i e é carregado na doca de saída h ,
 $q_i \geq 0$: número de paletes no veículo de entrega quando ele chega ao cliente i ,
 $\pi_i^p \geq 0$: número de produtos p no veículo de entrega quando ele chega ao cliente i ,

- $t_i \geq 0$: instante de chegada ao cliente i ,
 $t_{N+1} \geq 0$: instante de chegada ao *cross-dock*,
 $et_i \geq 0$ ($lt_i \geq 0$): tempo de antecipação (atraso) na entrega da demanda ao cliente i .

Considerando que os índices 0 e $N + 1$ representam o *cross-dock*, o **Modelo M1** é apresentado a seguir.

$$\min \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{\substack{j \in \mathcal{N} \cup \{N+1\} \\ |i \neq j}} CT_{ij} x_{ij} + \sum_{i \in \mathcal{N}} CEet_i + \sum_{i \in \mathcal{N}} CLlt_i + COdt^{max} \quad (3.1)$$

Sujeito a:

// Designação e ordenação das cargas de entrada

$$\sum_{f \in \mathcal{F}} z_{lf}^{in} = 1 \quad \forall l \in \mathcal{L} \quad (3.2)$$

$$w_{lm}^{in} + w_{ml}^{in} \geq z_{lf}^{in} + z_{mf}^{in} - 1 \quad \forall l, m \in \mathcal{L} (l < m); f \in \mathcal{F} \quad (3.3)$$

$$ut_l \geq PT_l \quad \forall l \in \mathcal{L} \quad (3.4)$$

$$ut_m \geq ut_l + PT_m - M(1 - w_{lm}^{in}) \quad \forall l, m \in \mathcal{L} (l \neq m) \quad (3.5)$$

// Designação e ordenação das cargas de saída

$$\sum_{h \in \mathcal{H}} z_{ih}^{out} = x_{0i} \quad \forall i \in \mathcal{N} \quad (3.6)$$

$$w_{ij}^{out} + w_{ji}^{out} \geq z_{ih}^{out} + z_{jh}^{out} - 1 \quad \forall i, j \in \mathcal{N} (i < j); h \in \mathcal{H} \quad (3.7)$$

$$dt_j \geq dt_i + CT + LT q_j - M(1 - w_{ij}^{out}) \quad \forall i, j \in \mathcal{N} (i \neq j) \quad (3.8)$$

$$dt^{max} \geq dt_i \quad \forall i \in \mathcal{N} \quad (3.9)$$

// Conexão entre as cargas e as rotas

$$\sum_{i \in \mathcal{N}} \sum_{h \in \mathcal{H}} \rho_{lfi}^p = R_l^p z_{lf}^{in} \quad \forall p \in \mathcal{P}; l \in \mathcal{L}; f \in \mathcal{F} \quad (3.10)$$

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{h \in \mathcal{H}} \rho_{lfi}^p \geq \pi_i^p - \left(\sum_{l \in \mathcal{L}} R_l^p \right) (1 - x_{0i}) \quad \forall p \in \mathcal{P}; i \in \mathcal{N} \quad (3.11)$$

$$\sum_{p \in \mathcal{P}} \sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \rho_{lfi}^p \leq \left(\sum_{p \in \mathcal{P}} \sum_{l \in \mathcal{L}} R_l^p \right) z_{ih}^{out} \quad \forall i \in \mathcal{N}; h \in \mathcal{H} \quad (3.12)$$

$$\sum_{p \in \mathcal{P}} \sum_{f \in \mathcal{F}} \sum_{h \in \mathcal{H}} \rho_{lfi}^p \leq \left(\sum_{p \in \mathcal{P}} R_l^p \right) y_{li} \quad \forall l \in \mathcal{L}; i \in \mathcal{N} \quad (3.13)$$

$$rt_i \geq ut_l + \sum_{p \in \mathcal{P}} \sum_{f \in \mathcal{F}} \sum_{h \in \mathcal{H}} TR_{fh}^p \rho_{lfi}^p - M(1 - y_{li}) \quad \forall l \in \mathcal{L}; i \in \mathcal{N} \quad (3.14)$$

$$dt_i \geq rt_i + LT q_i - M(1 - x_{0i}) \quad \forall i \in \mathcal{N} \quad (3.15)$$

$$t_i \geq dt_i + TT_{0i} - M(1 - x_{0i}) \quad \forall i \in \mathcal{N} \quad (3.16)$$

// Definição das rotas

$$\sum_{\substack{i \in \mathcal{N} \cup \{0\} \\ i \neq j}} x_{ij} = 1 \quad \forall j \in \mathcal{N} \quad (3.17)$$

$$\sum_{\substack{j \in \mathcal{N} \cup \{N+1\} \\ i \neq j}} x_{ij} = 1 \quad \forall i \in \mathcal{N} \quad (3.18)$$

$$q_j \geq Q_j + q_i - CAP(1 - x_{ji}) \quad \forall i, j \in \mathcal{N} (j \neq i) \quad (3.19)$$

$$Q_i \leq q_i \leq CAP \quad \forall i \in \mathcal{N} \quad (3.20)$$

$$\pi_j^p \geq D_j^p + \pi_i^p - \left(\sum_{l \in \mathcal{L}} R_l^p \right) (1 - x_{ji}) \quad \forall p \in \mathcal{P}; \forall i, j \in \mathcal{N} (j \neq i) \quad (3.21)$$

$$\pi_i^p \geq D_i^p \quad \forall p \in \mathcal{P}; \forall i \in \mathcal{N} \quad (3.22)$$

$$t_j \geq t_i + ST_i + TT_{ij} - M(1 - x_{ij}) \quad \forall i \in \mathcal{N}, j \in \mathcal{N} \cup \{N+1\} (i \neq j) \quad (3.23)$$

$$A_i - et_i \leq t_i \leq B_i + lt_i \quad \forall i \in \mathcal{N} \quad (3.24)$$

$$t_{N+1} \leq T^{max} \quad (3.25)$$

// Domínio das variáveis

A função objetivo (3.1) minimiza a soma dos custos de transporte, a soma dos custos devidos às violações das janelas de tempo dos clientes e o custo operacional do *cross-dock* dado em função do *makespan*.

As decisões de designação e de ordenação no descarregamento das cargas de entrada são modeladas pelas restrições (3.2)-(3.5). As restrições (3.2) garantem que cada carga de entrada seja designada para somente uma doca de entrada. As restrições (3.3) e (3.5) determinam a relação de precedência entre todas as cargas designadas para uma mesma doca de entrada, de forma que, se $z_{lf}^{in} = z_{mf}^{in} = 1$, então uma das variáveis w_{lm}^{in} e w_{ml}^{in} deverá ser igual a 1. O instante de encerramento do processamento de cada carga de entrada (ut_l) é definido pelas restrições (3.4) para as primeiras cargas processadas e pelas restrições (3.5) para as outras cargas de acordo com a ordem dos processamentos definida.

De forma análoga, temos as decisões de designação e de ordenação no carregamento das cargas de saída modeladas pelas restrições (3.6)-(3.9). Neste caso, cada carga de saída está associada a uma rota e é indicada pelo índice do primeiro cliente visitado. Na Figura 9, ilustramos o funcionamento do *cross-dock* em um determinado instante. Observe, na figura, que duas cargas de entradas estão sendo processadas, uma carga de saída cuja rota inicia no cliente 3 está carregada no veículo de entrega e pronto para partir e realizar as entregas e outra carga de saída cuja rota inicia com o cliente 7 ainda está sendo consolidada. No caso da figura, as variáveis x_{03} e x_{07} têm valores iguais a 1 e, para simplificar a escrita, denominaremos as duas cargas de saída pelos índices correspondentes aos primeiros clientes visitados, por exemplo, carga de saída 3 e carga de saída 7. Sempre que x_{0i} assumir o valor 1, as restrições

(3.6) determinam que a carga de saída i deve ser designada para uma das docas de saída e a ordenação do carregamento é definida pelas restrições (3.7), de forma similar às restrições (3.3). Os instantes de encerramento das operações de carregamento das cargas de saída são definidos pelas restrições (3.8), lembrando que existe uma conexão desses instantes com os tempos de liberação (rt_i) estabelecida pelas restrições (3.15) que são detalhadas no próximo parágrafo. Como o *makespan* (dt^{max}) é penalizado na função objetivo, as restrições (3.9) impõem que ele seja igual ao instante de encerramento do carregamento da última carga de saída processada.

Figura 9 – Ilustração do funcionamento de um *cross-dock* em um determinado instante.

Fonte: Elaborada pelo autor.

O conjunto de restrições (3.10)-(3.16) relacionam os instantes de disponibilidade das cargas de saída (rt_i) com os instantes de término dos processamentos das cargas de entrada (ut_l) e das cargas de saída (dt_i), considerando o tempo de transferência interna (TR_{fh}^p) e o número de produtos transferidos (ρ_{lfi}^p). As restrições (3.10), (3.11) e (3.12) garantem que o número total de produtos de cada tipo recebido seja suficiente para satisfazer as demandas dos clientes que têm seus pedidos consolidados em uma mesma carga de saída (i.e, clientes que estão em uma mesma rota), considerando as docas de origem e de destino. Como o número de produtos demandados é exatamente o número de produtos a serem distribuídos, $\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{h \in \mathcal{H}} \rho_{lfi}^p$ assume o valor mínimo. As restrições (3.13) indicam a existência de transferência de produtos entre as cargas de entrada l e a carga de saída i forçando a variável y_{li} assumir 1. Assim, quando houver transferência de uma carga de entrada para uma carga de saída, as restrições (3.14) impõem a condição de que a disponibilidade da carga de saída deve respeitar o tempo de término do processamento das cargas de entrada das quais ela depende e do tempo necessário para a transferência dos produtos. Por exemplo, na Figura 9, a carga de saída 7 não pode ser carregada, pois faltam produtos para compor os pedidos que compõem essa carga. Esses produtos não estão disponíveis porque a carga de entrada 1 ainda está sendo processada. As restrições (3.15) impõem o instante de encerramento do carregamento de uma carga de saída que deve respeitar

a disponibilidade da carga e o tempo para transferir os paletes consolidados para o interior do veículo. Já as restrições (3.16) estabelecem a conexão entre o encerramento do carregamento das cargas de saída com os instantes de chegada nos primeiros clientes visitados.

Na parte do roteamento, as restrições (3.17) e (3.18) asseguram que cada cliente é visitado exatamente uma vez. As restrições (3.19) determinam o número de paletes no veículo de entrega quando ele chega no cliente i (q_i é igual ao número de paletes do pedido do cliente i mais o número de paletes dos pedidos dos próximos clientes da rota). Os limitantes inferior e superior para a variável q_i são definidos pelas restrições (3.20), dessa forma, o número de paletes no veículo não pode ultrapassar a capacidade no início da rota e não pode ser menor que o número de paletes demandados pelo último cliente de cada rota. As restrições (3.21) e (3.22) definem o número de produtos do tipo p no veículo ao chegar no cliente i , de forma análoga às restrições (3.19) e (3.20) no caso do número de paletes. A sucessão dos tempos de chegada nos clientes é definida pelas restrições (3.24) de acordo com a ordem das visitas. As restrições (3.23) estabelecem as janelas de tempo de cada cliente permitindo a antecipação ou o atraso das entregas. Finalmente, a restrição (3.25) impõem o tempo máximo para o retorno do veículos de entrega ao *cross-dock*. O domínio das variáveis está definido anteriormente nas suas descrições.

3.2.2 Modelo M2

A eficiência dos métodos para resolver problemas de programação inteira pode ser afetada pelo modelo desenvolvido para representar o problema. Teoricamente, modelos com melhor relaxação linear pode levar a uma melhor performance de métodos exatos. A seguir, apresentamos uma formulação alternativa para o problema estudado. Incluímos variáveis binárias que indicam a rota a que cada cliente pertence ($v_{ij} \in \{0, 1\}$ em que v_{ij} é igual a 1 se o cliente i pertence a rota que inicia com a entrega do cliente j , e 0, caso contrário). As variáveis v_{ij} são definidas apenas para os índices do conjunto de clientes $\mathcal{N}$ ($i \neq j$). Nos modelos apresentados em Agustina, Lee e Piplani (2014) e em Rahbari *et al.* (2019), aparecem variáveis similares a essas, porém, como os autores utilizam a formulação de roteamento com três índices, elas indicam o veículo responsável pela entrega de cada cliente.

Com a adição das variáveis v_{ij} , o Modelo M1 pode ser reescrito como definido a seguir. Algumas restrições podem ser eliminadas, mas é necessário incluir ou reescrever outras. É possível escrever o número total de paletes e dos diferentes tipos de produtos através de expressões lineares (somatórios), portanto, as variáveis q_i e π_i^p e as restrições (3.19)-(3.22) podem ser eliminadas. Aplicadas essas modificações, temos o **Modelo M2** apresentado a seguir.

$$\min \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{\substack{j \in \mathcal{N} \cup \{N+1\} \\ i \neq j}} CT_{ij}x_{ij} + \sum_{i \in \mathcal{N}} CEet_i + \sum_{i \in \mathcal{N}} CLlt_i + COdt^{max}$$

Sujeito a:

// Designação e ordenação das cargas de entrada

(3.2) – (3.5)

// Designação e ordenação das cargas de saída

(3.6), (3.7), (3.9)

$$dt_j \geq dt_i + CT + LT(Q_j x_{0j} + \sum_{n \in N | n \neq j} Q_n v_{nj}) - M(1 - w_{ij}^{out}) \quad \forall i, j \in \mathcal{N} (i \neq j) \quad (3.26)$$

// Conexão entre as cargas e as rotas

(3.10), (3.12) – (3.14), (3.16)

$$\sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{h \in \mathcal{H}} \rho_{lfi}^p = D_i^p x_{0i} + \sum_{j \in N | j \neq i} D_j^p v_{ji} \quad \forall p \in \mathcal{P}; i \in \mathcal{N} \quad (3.27)$$

$$dt_i \geq rt_i + LT(Q_i x_{0i} + \sum_{j \in N | j \neq i} Q_j v_{ji}) - M(1 - x_{0i}) \quad \forall i \in \mathcal{N} \quad (3.28)$$

// Definição das rotas

(3.17), (3.18), (3.23) – (3.25)

$$v_{ij} \geq x_{0j} - (1 - x_{ji}) \quad \forall i, j \in \mathcal{N} (i \neq j) \quad (3.29)$$

$$v_{ij} \geq v_{nj} - (1 - x_{in}) \quad \forall i, j, n \in \mathcal{N} (i \neq j, i \neq n, j \neq n) \quad (3.30)$$

$$\sum_{j \in \mathcal{N} | j \neq i} v_{ij} + x_{0i} = 1 \quad \forall i \in \mathcal{N} \quad (3.31)$$

$$\sum_{i \in \mathcal{N} | i \neq j} Q_i v_{ij} + Q_j x_{0j} \leq CAP x_{0j} \quad \forall j \in \mathcal{N} \quad (3.32)$$

// Domínio das variáveis

Como $q_i = Q_j x_{0j} + \sum_{n \in \mathcal{N} | n \neq j} Q_n v_{nj}$, obtemos as restrições (3.26) e (3.28) que são similares às restrições (3.8) e (3.15), respectivamente, do Modelo M1. As restrições (3.27) garantem as transferências dos produtos necessários para compor os pedidos dos clientes consolidados em uma mesma carga de saída. As restrições (3.29) e (3.30), juntamente com as restrições (3.31) e (3.32), impõem que a variável v_{ij} seja igual a 1 se o cliente i está na rota que inicia com a entrega do cliente j . As restrições (3.31) asseguram que cada cliente é designado para apenas uma rota caso ele não seja o primeiro a ser visitado. As restrições (3.32), além de não permitirem designações para rotas não existentes, impõem que a capacidade do veículo seja respeitada.

3.2.3 Considerações

Com a inclusão das variáveis v_{ij} , o Modelo M2 passa a ter um número maior de variáveis inteiras e um número menor de variáveis contínuas se comparado ao Modelo M1. Já em relação ao número de restrições, para obter o Modelo M2, retiramos um número de restrições da

ordem de $N^2 \times P$, porém incluímos um número da ordem de N^3 . Em termos de números de variáveis e de restrições, não temos diferenças muito significativas entre os modelos. Nos testes computacionais avaliaremos a performance dos modelos ao resolver as instâncias utilizando um *solver* de otimização. A seguir, listamos outras considerações em relação ao problema e às modelagens propostas.

3.2.3.1 Consolidação das cargas e veículos de entrega

Embora, no problema estudado, não sejam considerados produtos intercambiáveis para a consolidação das cargas, os modelos apresentados também podem ser aplicados aos casos que se tem esse tipo de produto. Além disso, conforme já mencionado, o número de veículos para realizar as entregas é uma decisão do problema. Porém, caso esse número seja limitado, devemos incluir a restrição $\sum_{i \in \mathcal{N}} x_{0i} \leq K$, em que K é o número de veículos disponíveis.

3.2.3.2 Função objetivo

Uma hipótese bastante comum nos problemas de roteamento de veículos é assumir os custos proporcionais aos tempos de viagem. Em nosso caso, se, além de consideramos essa hipótese, também considerarmos os custos pelas violações e o custo operacional proporcionais aos seus respectivos tempos e que a unidade de custo por unidade de tempo é a mesma para todos os casos, os modelos podem ser escritos de forma equivalente com a substituição da função objetivo (3.1) pela função objetivo (3.33).

$$\min \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{j \in \mathcal{N} \cup \{N+1\} | i \neq j} TT_{ij}x_{ij} + \sum_{i \in \mathcal{N}} et_i + \sum_{i \in \mathcal{N}} lt_i + dt^{max} \quad (3.33)$$

Dado que o foco do problema é essencialmente operacional, considerar os tempos para as transferências dos produtos das docas de entrada para as docas de saída podem contribuir para decisões mais precisas e, conseqüentemente, mais próximas a problemas reais. Além disso, o tempo para transferir os produtos pode representar mão-de-obra ou custos com utilização de equipamentos. Neste caso, se o objetivo for também minimizar esses custos, é possível incorporá-los aos modelos adicionando a expressão (3.34) na função objetivo.

$$\sum_{p \in \mathcal{P}} \sum_{l \in \mathcal{L}} \sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{N}} \sum_{h \in \mathcal{H}} TR_{fh}^p \rho_{lfi}^p \quad (3.34)$$

3.2.3.3 Limite inferior para instante de término do carregamento

Considerando que todas as entregas devem ser realizadas, é possível calcular um limite inferior para a variável dt_i . Seja $\bar{L}_i = \{l \in \mathcal{L} : \exists p \in \mathcal{P} | R_l^p > 0 \wedge D_i^p > 0\}$, i.e., o conjunto de todas as cargas de entrada com pelo menos um tipo de produto necessário para compor o pedido do cliente i ; $\alpha_i^* = \operatorname{argmax}_{l \in \bar{L}_i} \{PT_l\}$, i.e., a carga com maior tempo de processamento entre as cargas

das quais o cliente i recebe algum tipo de produto; e $\tau_i^* = \min_{f \in \mathcal{F}, h \in \mathcal{H}} \left\{ \sum_{p \in \mathcal{P}: R_{\alpha_i^*}^p > 0} TR_{fh}^p D_i^p \right\}$, i.e.,

o tempo mínimo para transferir da área de recebimento para a área de consolidação os produtos demandados pelo cliente i que estão na carga de entrada α_i^* . Supondo que cada pedido fosse enviado como uma entrega direta e que também fosse o primeiro a ser despachado, um limite inferior para o término do carregamento (dt_i) é o tempo de processamento da carga de entrada α_i^* somado ao tempo mínimo de transferência e ao tempo de carregamento, como escrito na inequação (3.35).

$$dt_i \geq PT_{\alpha_i^*} + \tau_i^* + Q_i LT \quad (3.35)$$

3.3 Experimentos computacionais

Os experimentos computacionais foram realizados em três fases. Primeiro, analisamos comparativamente a performance dos Modelos M1 e M2. Na sequência, apresentamos uma comparação em termos de modelagem do modelo apresentado neste trabalho com um modelo da literatura para o VRCDSP. Por último, avaliamos a relevância de integrar a programação do *cross-dock* com o problema de roteamento de veículo para obter um plano de distribuição comparado às estratégias hierárquicas e de entrega direta.

As instâncias foram geradas a partir de uma adaptação de instâncias clássicas para o problema de roteamento de veículos. Para resolver as instâncias, foi utilizado o *solver* comercial Gurobi versão 8.1 na configuração padrão e tempo limite de 2 horas em um computador Intel Core i7-2600 2.40Gz com 15.6 GB de memória RAM no sistema operacional Linux 18.04 LTS.

3.3.1 Geração das instâncias

Na literatura, não encontramos instâncias específicas para o problema estudado. Dessa forma, adaptamos as instâncias R101-R112 (SOLOMON, 1987) selecionando os dados referentes a seus N primeiros clientes. Essas instâncias têm as posições dos clientes geradas randomicamente que é uma configuração mais próxima do caso real no qual este trabalho foi baseado. Para cada instância, são mantidas as distâncias entre os clientes e entre os clientes e o depósito (*cross-dock*). Em nosso caso, existe o tempo de operação no *cross-dock* para as atividades de descarregamento, de consolidação e de carregamento antes do início das rotas de entrega, por isso, postergamos as aberturas e os fechamentos das janelas de tempo adicionando a média do tempo de processamento das cargas de entrada por doca (Γ). Assim, a janela de tempo de cada cliente i ($[A_i, B_i]$) e o tempo máximo T^{max} são obtidos da seguinte forma:

$$\Gamma = \left\lceil \sum_{l \in \mathcal{L}} IP_l / F \right\rceil,$$

$$[A_i, B_i] = [A_i^o + \Gamma, B_i^o + \Gamma]$$

e

$$T^{max} = T^{max^o} + \Gamma$$

onde A_i^o , B_i^o e T^{max^o} são os valores nas instâncias originais.

Assim, preservamos os intervalos das janelas de tempo que são distintos. As instâncias R101-R104, R105-R106 e R109-R112 têm janelas de tempos com intervalos de 10, 30 e 65 unidades de tempo, respectivamente. Além disso, para cada caso, o número de clientes cujas janelas de tempo apresentam esses intervalos é decrescente. Por exemplo, no caso de 25 clientes, R101, R102, R103 e R104 têm, respectivamente, 25, 14, 12 e 8 clientes com janelas de tempo de intervalos iguais a 10, e os restantes apresentam janelas de tempo mais largas. Na Tabela 6, definimos outros parâmetros considerados na geração dos três conjuntos de instâncias de acordo com o número de clientes.

Tabela 6 – Parâmetros de cada conjunto de instâncias de acordo com o número de clientes.

	15 Clientes	25 Clientes	40 Clientes
Cargas de entrada (L)	5	10	15
Docas de entrada (F)	3	4	6
Docas de saída (H)	3	4	6
Tipos de produto (P)	10	15	25

Fonte: Elaborada pelo autor.

Uma parte dos dados é obtida aleatoriamente considerando dois cenários e, para cada cenário, três grupos conforme indicado na Tabela 7 com os intervalos para a geração uniforme. No Cenário 1, o número de paletes Q_i necessário para organizar cada pedido é definido entre 25% e 100% da capacidade do veículo e, no Cenário 2, esse número é definido entre 12% e 50% da capacidade do veículo. Consequentemente, é possível ter rotas com mais clientes no segundo caso. Para cada cenário, geramos três grupos de instâncias baseados no número de tipos de produtos (η_i) demandados pelo cliente i . Esse número é obtido aleatoriamente considerando as faixas de 1 a 20% P , de 1 a 40% P e de 1 a 100% P , configurando os Grupos A, B e C, respectivamente. As matrizes $(I_i^p) \in \{0, 1\}^{\mathcal{P} \times \mathcal{N}}$ e $(J_i^p) \in \{0, 1\}^{\mathcal{P} \times \mathcal{L}}$ indicam quais tipos de produtos são demandados pelos clientes e em quais cargas de entrada eles estão, respectivamente. (I_i^p) é aleatoriamente definida respeitando o número de tipos de produtos demandados pelo cliente e garantido que cada tipo de produto deve ser demandado por pelo menos um cliente. (J_i^p) é aleatoriamente obtida garantindo que cada tipo de produto está em apenas uma das cargas de entrada e cada carga de entrada tem pelo menos um tipo de produto e não mais do que três. O volume de cada tipo de produto (Θ^p) é gerado randomicamente entre 1% e 10% do volume suportado por um paleta.

Outra parte dos dados é obtida através de cálculos com os parâmetros e com os valores obtidos aleatoriamente, como reportados na Tabela 8. Primeiramente, para cada cliente i , o

Tabela 7 – Dados gerados aleatoriamente.

$Q_i \in \mathbb{Z}$	número de paletes para organizar o pedido do cliente i Cenário 1: $U[0, 25CAP; CAP]$ – Cenário 2: $U[0, 125CAP; 0, 5CAP]$
$\eta_i \in \mathbb{Z}$	número de tipos de produtos demandados pelo cliente i Grupo A: $U[1; 0, 2P]$ – Grupo B: $U[1; 0, 4P]$ – Grupo C: $U[1, P]$
$I_i^p \in \{0, 1\}$	assume 1, se o produto p é demandado pelo cliente i , e 0, caso contrário
$J_l^p \in \{0, 1\}$	assume 1, se o produto p está na carga de entrada l , e 0, caso contrário
$\Theta^p \in \mathbb{R}$	volume de uma unidade do produto p $U[0, 01VOL; 0, 1VOL]$

Fonte: Elaborada pelo autor.

volume total de cada produto p (V_i^p) demandado é o piso do valor obtido da divisão do volume que corresponde ao total de paletes (Q_iVOL) pelo número de tipos de produtos (η_i) necessários para compor o pedido do cliente i . O número de cada tipo de produto (D_i^p) relacionado no pedido de cada cliente é o piso da divisão de V_i^p por θ^p . Com as demandas de cada cliente, contabilizamos o número de cada tipo de produto em cada carga de entrada (R_l^p) e o tempo de processamento (IP_l), considerando o volume total da carga e um tempo de 5 unidades para processar o volume correspondente ao volume de um paleta (VOL). A distância entre duas docas ($DIST_{fh}$) é calculada de acordo com as suas posições que são usadas para definir o tempo de transferência para cada tipo de produto (TR_{fh}^p), que depende da velocidade de transferência (0,4) e é proporcionalmente ao volume do produto em relação ao volume suportado por um paleta.

Tabela 8 – Dados calculados.

V_i^p	volume total aproximado do produto p demandado pelo cliente i	$V_i^p = I_i^p \left\lfloor \frac{Q_iVOL}{\eta_i} \right\rfloor$
D_i^p	número de produtos p demandados pelo cliente i	$D_i^p = \left\lfloor \frac{V_i^p}{\Theta^p} \right\rfloor$
R_l^p	número de produtos p na carga de entrada l	$R_l^p = J_l^p \sum_{i \in \mathcal{N}} D_i^p$
IP_l	tempo de processamento da carga de entrada l	$IP_l = 5 \sum_{p \in \mathcal{P}} \left\lceil \frac{R_l^p \Theta^p}{VOL} \right\rceil$
$DIST_{fh}$	distância entre a doca de entrada f e a doca de saída h	$DIST_{fh} = 10 + 2 f - h $
TR_{fh}^p	tempo para transferir uma unidade do produto p de acordo com as docas f e h	$TR_{fh}^p = 0,4 \frac{\Theta^p}{VOL} DIST_{fh}$

Fonte: Elaborada pelo autor.

Por último, definimos o para tempo de transferência de um paleta da área de estoque temporário para o interior do veículo, tempo de troca dos veículos e o tempo de serviço em cada

cliente, $CT = LT = 2$ e $ST_i = 2Q_i$. Além disso, assumimos a proporcionalidade entre os custos e tempos e um custo igual por unidade de tempo nos três casos (tempo de viagem, *makespan*, tempo de violação). Assim, utilizamos a função objetivo (3.33) que corresponde à minimização do tempo total de viagem somado com tempo total de violação das janelas de tempo e com o *makespan*. Cada instância gerada é identificada da seguinte maneira: nome da instância de Solomon - cenário - grupo. Por exemplo, a instância R111-2B corresponde a instância gerada a partir da instância de Solomon R111 considerando o Cenário 2 e o Grupo B.

3.3.2 Fase 1 - Análise dos modelos desenvolvidos

Para analisar os dois modelos desenvolvidos, usamos os conjuntos de instâncias com 15 e 25 clientes. A seguir, apresentamos uma compilação dos resultados. Detalhes sobre as soluções obtidas para as instâncias estão disponíveis no Apêndice A.

• Relaxação linear

Primeiramente, resolvemos a relaxação linear para todas as instâncias com e sem os limitantes inferiores para as variáveis dt_i , como definido na Subseção 3.2.3. Com a inclusão dos limitantes inferiores, o valor da função objetivo da relaxação linear aumentou, em média, 85% para o Modelo M1 e 51% para o Modelo M2. A seguir, comparamos as relaxações lineares dos dois modelos com a inclusão dos limitantes inferiores para as variáveis dt_i .

Figura 10 – Relaxação linear (M1 × M2).

Fonte: Elaborada pelo autor.

Na Figura 10, ilustramos os gráficos com os valores obtidos da função objetivo das relaxações lineares dos dois modelos. Os maiores valores são obtidos pelo Modelo M2. Para as

instâncias com 15 clientes, em média, os valores obtidos pelo M2 são aproximadamente 28% (Cenário 1) e 16% (Cenário 2) maiores que os valores obtidos pelo M1. A maior e a menor diferença são, respectivamente, de 33,6% e de 20,6%, para o Cenário 1, e de 21,9% e de 12,7%, para o Cenário 2. No caso das instâncias com 25 clientes, os maiores valores também são obtidos pelo Modelo M2 e com diferenças de 60% (Cenário 1) e 30% (Cenário 2), em média, sendo que a maior e a menor diferença são, respectivamente, de 72,6% e de 49,6% para o Cenário 1 e de 40,1% e de 25,79% para o Cenário 2.

• Performance dos modelos

Para comparar a performance do *solver* ao resolver as instâncias com os dois modelos desenvolvidos, dividimos as instâncias em dois grupos de acordo com o número de clientes. No caso das instâncias com 15 clientes, quando obtemos soluções ótimas para 52 instâncias das 72, analisamos os tempos de solução. Na Figura 11, reportamos o número acumulativo de instâncias para as quais foram obtidas soluções ótimas ao longo de 2 horas para os dois modelos. Observamos que o Modelo M2 tem uma pequena vantagem sobre o Modelo M1 apresentando soluções ótimas para três instâncias a mais.

Figura 11 – Número de soluções ótimas obtidas no decorrer de 2 horas (M1 × M2) - 15 clientes.

Fonte: Elaborada pelo autor.

No caso das instâncias com 25 clientes, apenas a instância R101-1A teve sua solução ótima encontrada e o tempo para obtê-la com M2 é 40% menor que o tempo gasto com M1. Conforme ilustrado na Figura 12, comparamos os limitantes duais e os primais obtidos no final de duas horas com cada um dos modelos. Em relação aos limitantes primais, os valores obtidos com os dois modelos são bastantes próximos com poucos casos nos quais, com o Modelo M2, são obtidos valores um pouco menores que os obtidos com o M1. Já na comparação dos limitantes duais, observamos que foram obtidos valores maiores com o Modelo M2, principalmente, no

Cenário 2. Consequentemente, as soluções obtidas com o Modelo M2 apresentam *gaps* menores para a maioria das instâncias.

Figura 12 – Limitantes duais e primais obtidos em 2 horas (M1 $\times$ M2) - 25 clientes.

Fonte: Elaborada pelo autor.

Com base nos resultados obtidos, verificamos uma pequena vantagem do Modelo M2. Dessa forma, para as próximas discussões, consideramos apenas o Modelo M2.

3.3.3 Fase 2 - Comparação com a literatura

Como já mencionado na revisão da literatura, [Agustina, Lee e Piplani \(2014\)](#) e [Rahbari et al. \(2019\)](#) são, no que é de conhecimento do autor, as duas únicas abordagens do VRCDSP, porém estão baseadas no contexto de distribuição de alimentos perecíveis que é diferente do problema aqui abordado. Os modelos apresentados nos dois trabalhos usam basicamente as mesmas restrições na formulação das decisões de designação e de ordenação das cargas e também das decisões de roteamento. A diferença é que [Rahbari et al. \(2019\)](#) considera uma rede de muitos para muitos e tempos distintos nos processamentos das cargas. Além disso, os autores consideram a maximização do frescor com que os produtos chegam ao seus destinos, tornando o problema multi-objetivo. Embora as características operacionais resultem em problemas diferentes, os modelos da literatura e o abordado têm em comum as decisões de designação e de ordenação de cargas e as de roteamento de veículos, porém modeladas de formas distintas. Assim, para uma comparação entre os modelos da literatura e o proposto, fizemos algumas adaptações de forma

que os VRSCDSPs representem um mesmo problema. Para isso, partimos do VRCDSP definido em Agustina, Lee e Piplani (2014).

No modelo apresentado por Agustina, Lee e Piplani (2014), não consideramos o tempo de estoque dos produtos dentro do *cross-dock*. Consequentemente, eliminamos o custo de estoque e as restrições a ele relacionadas. No Modelo M2, tomamos os custos e os tempos de processamento como definidos em Agustina, Lee e Piplani (2014) e fixamos o número de veículos (K). Além disso, ignoramos os tempos para as transferências interna, o *makespan* e os tempos de serviço. Para estabelecer o caso de distribuição um para um, assumimos que cada cliente demanda apenas um tipo de produto e cada tipo de produto é demandado por apenas um cliente, assim temos o número de produtos igual ao número de clientes. Com essas considerações, os dois modelos se tornam equivalentes.

Como as instâncias utilizadas em Agustina, Lee e Piplani (2014) não estão disponíveis, geramos oito conjuntos de instâncias baseados nos parâmetros das instâncias utilizadas pelos autores. O número de docas de entrada (2), o número de cargas de entrada (L), o número de veículos de entregas (K), o número de pedidos/clientes (N), os tempos de processamento e os custos são definidos como em Agustina, Lee e Piplani (2014). As demandas foram randomicamente geradas entre 2 e 8 paletes. Cada conjunto de instâncias foi adaptado a partir das instâncias R201-R205 (SOLOMON, 1987), escolhidas por considerarem um cenário em que muitos clientes podem ser servidos por um mesmo veículo (AGUSTINA; LEE; PIPLANI, 2014).

Tabela 9 – Comparação dos modelos adaptados.

Conjunto de instâncias	Agustina, Lee e Piplani (2014)			Modelo M2		
	SO	T(s)	GAP(%)	SO	T(s)	GAP(%)
2/2/10	5	<1	0,00	5	<1	0,00
2/2/20	5	34	0,00	5	25	0,00
2/2/30	4	2980	1,11	5	2055	0,00
2/2/40	0	7200	17,61	0	7200	11,62
4/4/10	5	<1	0,00	5	<1	0,00
4/4/20	5	297	0,00	5	35	0,00
4/4/30	5	2494	0,00	5	469	0,00
4/4/40	0	7200	9,38	3	4650	3,46
Média	3,63	2527	3,51	4,13	1804	1,89

L: número de cargas de entrada K: número de veículos N: número de clientes SO: número de soluções ótimas

GAP: *gap* T: tempo médio de solução

Fonte: Elaborada pelo autor.

Resolvemos as instâncias usando o *solver* Gurobi 8.1 na configuração padrão com o mesmo computador e tempo limite de 2 horas. Na Tabela 9, reportamos, para cada conjunto de instâncias, o número de soluções ótimas obtidas (SO), o tempo médio de solução em segundos

(T) e o gap^2 médio (GAP) para os dois modelos. Em geral, o Modelo M2 teve uma melhor performance em relação ao modelo da literatura. Essa vantagem deve ser resultante do uso de variáveis de dois índices na modelagem das decisões de roteamento de veículos, que elimina alguns casos de simetrias nas soluções. Esses resultados indicam que os modelos de [Agustina, Lee e Piplani \(2014\)](#) e [Rahbari et al. \(2019\)](#) podem ser melhorados com uma reformulação usando as variáveis de dois índices na parte do roteamento.

3.3.4 Fase 3 - Relevância da integração dos problemas

Nesta fase dos experimentos computacionais, avaliamos as soluções obtidas pela estratégia integrada (EI), usando o Modelo M2, comparadas com as soluções obtidas utilizando as três estratégias frequentemente adotadas na prática para obter uma programação para um sistema de distribuição. Em uma delas, admitimos apenas entregas diretas para os consumidores (DD). Nas outras duas, resolvemos hierarquicamente os problemas de programação do *cross-dock* e de roteamento de veículos. Na estratégia *cross-dock*/roteamento (CDR), a programação para as atividades no *cross-dock* são determinadas antes da definição das rotas de entregas. Na estratégia roteamento/*cross-dock* (RCD), definimos as rotas de entrega primeiro e, então, obtemos uma programação para as operações no *cross-dock* respeitando as cargas de saída definidas de acordo as rotas estabelecidas. Para cada estratégia, resolvemos as instâncias dos três conjuntos (15, 25 e 40 clientes) usando o Modelo M2 com as seguintes adaptações:

- DD: uma rota para cada cliente é estabelecida fixando as variáveis de roteamento ($x_{0i} = x_{i,N+1} = 1$) e as restrições que se tornam redundantes são eliminadas.
- CDR: primeiramente, resolvemos o problema de programação do *cross-dock* admitindo apenas entregas diretas como na estratégia DD. Com a solução obtida, definimos as operações do descarregamento e em quais docas de saída cada cliente é servido. Na sequência, resolvemos o problema de roteamento para cada doca de saída permitindo consolidações entre os clientes servidos numa mesma doca e estabelecendo a ordem dos carregamentos das cargas de saída consolidadas.
- RCD: primeiro, resolvemos o problema de roteamento impondo o tempo de término do carregamento de todas as cargas é maior ou igual a Γ (definido na Subseção 3.3.1) para simular o tempo de operação para a consolidação e o carregamento das cargas de saída no *cross-dock*. Depois, resolvemos o problema de programação do *cross-dock* com as carga de saídas definidas de acordo com as rotas estabelecidas.

No caso das estratégias hierárquicas, o tempo para resolver cada um dos subproblemas é limitado a uma hora. Na sequência, apresentamos uma análise dos resultados para cada um dos

² No caso do *solver* Gurobi, o gap fornecido é a diferença percentual dos limitantes primal e dual em relação ao limitante primal.

conjuntos de instâncias. Mais detalhes sobre as soluções obtidas estão disponíveis no Apêndice A.

3.3.4.1 Resultados para as instâncias com 15 clientes

Como já mencionado anteriormente, obtemos soluções ótimas para 52 de um total de 72 instâncias com a EI. Com a estratégia DD, tivemos soluções ótimas para a maioria das instâncias, apenas 8 a solução ótima não foi atingida (os *gaps* foram menores que 2%). Consequentemente, o mesmo acontece com a CDR quando o problema de programação do *cross-dock* é resolvido primeiro. Nestes casos, a não otimalidade não altera nossas conclusões já que as soluções obtidas com a EI são significativamente melhores que as soluções obtidas com a DD e a CDR. Já com a estratégia RCD, os dois problemas foram resolvidos até a otimalidade.

Figura 13 – Comparação entre as quatro estratégias para as instâncias com 15 clientes.

Fonte: Elaborada pelo autor.

Na Figura 13, o gráfico mostra os valores da função objetivo de acordo com cada estratégia. Como esperado, as soluções fornecidas pelas estratégias DD, CDR e RCD apresentam valores da função objetivo sempre maiores ou iguais aos valores associados às soluções ótimas obtidas com a EI. Em particular, mesmo no Cenário 1 que tem clientes com demandas mais próximas da capacidade dos veículos de entrega, as piores soluções são obtidas predominantemente

com a estratégia de entrega direta (DD) o que reforça a importância do roteamento para o sistema de distribuição. Em relação aos valores da função objetivo, as melhores performances são das estratégias EI e RCD, embora não sendo possível obter soluções para as instâncias R110-1A, R111-1B, R101-2B e R109-2B utilizando a RCD, uma vez que as rotas definidas levaram a impossibilidade de consolidação no *cross-dock*, i.e., não é possível completar a rota definida e retornar ao *cross-dock* antes do tempo limite para algumas cargas de saída. Além disso, para nove instâncias, a RCD fornece as piores soluções. De acordo com esses resultados, podemos concluir que a integração dos problemas de programação de *cross-dock* e de roteamento é relevante para obter um plano de distribuição com um menor custo.

Tabela 10 – Performance da estratégia EI em relação à RCD - 15 clientes.

Cenário	Grupo	NI	$\bar{D}$	D^{min}	D^{max}	SO	GAP (%)
1	A	10	4,58	0,03	22,79	12	0,00
	B	11	4,11	0,01	9,98	10	2,29
	C	11	5,72	1,22	15,66	8	6,56
2	A	11	9,48	1,86	18,43	10	10,01
	B	11	11,12	3,40	24,59	6	11,28
	C	11	8,34	1,34	20,33	6	8,73
Média			7,23	1,31	18,63		6,58

NI: número de casos em que EI foi melhor $\bar{D}$: diferença percentual média entre as soluções

D^{min} : menor diferença percentual D^{max} : maior diferença percentual

SO: número de soluções ótimas GAP: *gap*

Fonte: Elaborada pelo autor.

Na Tabela 10, resumimos a análise da performance da estratégia EI em relação à RCD. Na tabela, são reportados o número de instâncias (NI) para as quais a solução com a EI foi melhor ou a RCD não encontrou solução factível. Também calculamos as diferenças percentuais (D) dos valores da função objetivo das melhores soluções e reportamos seu valor médio ($\bar{D}$), o maior valor (D^{max}) e o menor valor (D^{min}) das diferenças percentuais. Os valores de D foram obtidos da seguinte forma:

$$D = \frac{FO^{RCD} - FO^{EI}}{FO^{RCD}} \times 100,$$

onde FO^{RCD} e FO^{EI} são os valores da função objetivo das soluções obtidas com as estratégias RCD e EI, respectivamente. Também são reportados o número de instâncias (SO) para as quais foram obtidas soluções ótimas com a EI, o *gap* médio (GAP) apenas considerando as instâncias com *gaps* não nulos. De forma geral, com as informações apresentadas, verificamos que as soluções obtidas com a EI são melhores para a maioria das instâncias, podendo alcançar diferenças significativas maiores que 20% o que pode representar uma redução significativa no custo de distribuição do sistema.

Para comparar a programação obtida pelas estratégias EI e RCD e avaliar os efeitos sobre as decisões ao integrar os dois problemas, analisamos os tempos que compõem a função objetivo

separadamente. Para ambos os cenários, tivemos conclusões muito semelhantes, sendo mais evidentes para o Cenário 2. Por isso, escolhemos o Cenário 2 para apresentar aqui, no entanto, os dados estão disponíveis no Apêndice A para o Cenário 1.

Figura 14 – Tempos de acordo com as estratégias EI e RCD - Cenário 2.

Fonte: Elaborada pelo autor.

Na Figura 14, temos o tempo operacional, o tempo total de viagem e o tempo total de atraso obtidos na solução de cada instância do Cenário 2 de acordo com cada uma das estratégias. Como as soluções não apresentam tempo de antecipação, não incluimos essa informação nos gráficos. Em geral, a diferença entre os valores da função objetivo são principalmente em consequência das diferenças entre os tempos de viagem e os tempos de atraso. Conforme pode

ser observado, quando resolvemos o problema de roteamento de forma independente, obtemos rotas com menor tempo total de viagem com ganhos de 9,4% em média. Entretanto, as soluções apresentam um alto tempo de atraso quando usamos as rotas definidas para a programação do *cross-dock*. Obviamente, isso acontece porque as decisões das rotas não levam em consideração o tempo para a consolidação das cargas de saída no *cross-dock*. Os planos de distribuição obtidos com a estratégia EI apresentam um tempo de atraso, aproximadamente, 87,7% menor em média. Além disso, note que os tempos operacionais (*makespan*) nas soluções obtidas com a estratégia RCD são iguais ou pouco maiores, aproximadamente 4,2% em média, se comparado aos tempos apresentados nas soluções obtidas com a EI.

Com a análise dos tempos separadamente, a vantagem de integrar os dois problemas para obter um plano de distribuição fica ainda mais evidente. Além de soluções com custos menores, observamos também a melhor qualidade das soluções que apresentam um tempo de atraso consideravelmente menor. Os resultados também confirmam a relação de interdependência entre as decisões para os diferentes problemas considerados. Assim, para esse tipo de sistema de distribuição, concluímos que é relevante considerar a integração dos problemas para obter um plano de distribuição mais eficiente.

3.3.4.2 Resultados para as instâncias com 25 clientes

No caso das instâncias com 25 clientes, ou seja, em que a dimensão do problema é maior, a maioria das soluções apresenta *gaps* não nulos. No caso das estratégias hierárquicas, pelo menos um dos subproblemas não é resolvido até obter *gap* nulo em quase todos os casos. Comparamos as quatro estratégias e verificamos, como ocorre para as instâncias com 15 clientes, que as estratégias DD e CDR fornecem as piores soluções. Mesmo com as possíveis melhorias das soluções dessas estratégias de acordo com os *gaps* apresentados, elas não são competitivas com as soluções obtidas pelas estratégias EI e RCD. As estratégias EI e RCD tiveram um desempenho semelhante como apresentamos a seguir.

Nas Tabelas 11 e 12, resumimos as comparações entre as estratégias EI e RCD. As informações são similares às apresentadas na Tabela 10, apenas incluímos o número de soluções não ótimas entre parênteses ao lado do valor médio dos *gaps*. No caso da estratégia RCD, reportamos também a média dos *gaps* para cada um dos subproblemas resolvidos (Problemas de roteamento e de programação do *cross-dock*).

Como é possível observar, as duas estratégias apresentaram a mesma performance em relação a melhor solução (50% cada). Entretanto, em média, as soluções obtidas com a estratégia EI apresenta valor da função objetivo 2,3% menor. Importante ressaltar que, em média, os *gaps* são maiores que os dois *gaps* (Roteamento e CD) apresentados na Tabela 12. Além disso, o número de instâncias com *gaps* não nulos na Tabela 11 é bem maior que na Tabela 12. Diante dessas duas últimas observações, temos uma chance maior de obter soluções ainda melhores com a estratégia EI.

Tabela 11 – Performance da estratégia EI em relação à RCD - 25 clientes.

Cenário	Grupo	NI	$\bar{D}$	D^{min}	D^{max}	GAP (%)
1	A	6	2,74	0,13	14,14	2,26 (11)
	B	6	0,79	0,04	1,71	5,59 (12)
	C	7	2,47	0,02	6,27	7,34 (12)
2	A	7	3,69	0,43	8,87	14,07 (12)
	B	5	1,96	0,25	3,64	19,22 (12)
	C	5	2,20	0,31	6,39	20,80 (12)
Média			2,30	0,20	6,84	11,55

NI: número de casos em que EI foi melhor $\bar{D}$: diferença percentual média D^{min} : menor diferença percentual D^{max} : maior diferença percentualGAP: *gap*

Fonte: Elaborada pelo autor.

Tabela 12 – Performance da estratégia RCD em relação à EI - 25 clientes.

Cenário	Grupo	NI	$\bar{D}$	D^{min}	D^{max}	GAP (%)	
						Roteamento	CD
1	A	6	0,22	0,01	0,52	0,00	1,81 (7)
	B	5	0,76	0,04	2,07	0,00	3,16 (11)
	C	6	0,37	0,03	1,02	0,00	6,19 (12)
2	A	5	0,99	0,16	1,78	8,91 (8)	0,00
	B	7	2,10	0,09	5,09	9,47 (9)	2,01 (1)
	C	5	2,20	0,15	4,15	7,11 (9)	1,96 (8)
Média			1,11	0,08	2,44	4,25	2,52

NI: número de casos em que EI foi melhor $\bar{D}$: diferença percentual média GAP: *gap* D^{min} : menor diferença percentual D^{max} : maior diferença percentual

Fonte: Elaborada pelo autor.

Na Tabela 12, é interessante observar que os valores médios dos *gaps* refletem as dificuldades em resolver os problemas de roteamento e de programação do *cross-dock*. Conforme a geração das instâncias, sabemos que, no Cenário 1, as rotas podem ter até quatro clientes e, no Cenário 2, até oito clientes. Dessa forma, a natureza combinatorial do problema de roteamento é mais extensiva no Cenário 2 do que no Cenário 1. O inverso ocorre no caso do problema de programação do *cross-dock*, já que, no Cenário 1, tem um número maior de cargas de saída o que reflete em maior complexidade nas decisões de designação e de ordenação para as operações de carregamento.

3.3.4.3 Resultados para as instâncias com 40 clientes

No caso das instâncias com 40 clientes, as estratégias EI e RCD também fornecem as melhores soluções. Porém, tivemos algumas instâncias para as quais não foi possível obter uma

solução viável no tempo de duas horas. No caso da estratégia EI, tivemos soluções viáveis para 63 das 72 instâncias com o Modelo M2 e, com a estratégia RCD, não obtivemos solução viável apenas para a instância R108-2A. No Cenário 1, ambas estratégias são competitivas. RCD é 2,9% (em média) melhor que EI, entretanto, no Cenário 2, a diferença é maior (12,7% em média). Esses resultados apontam limitações para obter soluções viáveis ou boas com a estratégia EI quando se trata de problemas com maiores dimensões. Como esperado, à medida que a dimensão do problema aumenta, a resolução do problema integrado de programação de *cross-dock* e roteamento de veículos se torna mais difícil.

3.3.4.4 Tempo de solução: EI $\times$ RCD

Na Tabela 13, reportamos as médias dos tempos de solução para cada estratégia (EI e RCD). Para cada conjunto de instância, de acordo com o número de clientes, os valores estão separados em Cenários 1 e 2 e nos Grupos A, B e C. No caso da estratégia RCD, apresentamos o tempo médio de solução ao resolver o problema de roteamento e o problema de programação do *cross-dock* (CD). Lembrando que, para cada problema, o tempo de solução é limitado a uma hora (3600 s).

Tabela 13 – Tempo médio de solução em segundos para obter as melhores soluções com as estratégias EI e RCD.

	Cenário	Grupo	EI	RCD		
				Roteamento	CD	Total
15 clientes	1	A	259	<1	3	3
		B	1460	<1	3	3
		C	382	<1	137	137
	2	A	3163	14	<1	14
		B	4582	11	<1	11
		C	5108	11	<1	11
25 clientes	1	A	6670	<1	2525	2525
		B	7200	<1	3600	3600
		C	7200	<1	3600	3600
	2	A	7200	3019	396	3415
		B	7200	3094	915	4009
		C	7200	3210	3182	6392
40 clientes	1	A	7200	1	3600	3601
		B	7200	1	3600	3601
		C	7200	1	3600	3601
	2	A	7200	3600	3294	6894
		B	7200	3600	3600	7200
		C	7200	3600	3600	7200

Fonte: Elaborada pelo autor.

De acordo com os valores na Tabela 13, observamos que, em geral, a estratégia RCD fornece soluções mais rapidamente que a estratégia EI. Porém, é importante lembrar que a solução obtida com a RCD pode não ser viável, i.e., com rotas impraticáveis dentro do tempo limite. Note também que, à medida que a dimensão do problema aumenta, os tempos de solução da RCD se aproximam dos tempos de solução da EI. Conforme esperado, independente da estratégia, à medida que o número de variáveis aumenta, temos também um aumento no tempo de resolução das instâncias. Importante ressaltar que mesmo que os dois problemas sejam resolvidos na otimalidade e a solução obtida represente um plano viável de distribuição, não podemos garantir que esse plano tem custo mínimo. Para garantir um plano de distribuição para o sistema com custo mínimo, é necessário resolver os problemas integrados.

3.4 Conclusões

Neste capítulo, apresentamos dois modelos de programação inteira mista para o VRCDSP no contexto de um sistema de distribuição típico de uma rede de varejo. A comparação entre os modelos indicaram que o modelo no qual foram incluídas as variáveis indicativas de rotas apresentou uma relaxação linear com soluções de melhor qualidade, principalmente, com a inclusão das restrições que determinam um limitante inferior para as variáveis dt_i . Além disso, quando utilizado o Modelo M2, as instâncias são resolvidas mais facilmente pelo *solver* de otimização. Embora o VRCDSP estudado seja diferente do VRCDSP abordado em [Agustina, Lee e Piplani \(2014\)](#), com algumas adaptações, foi possível comparar o Modelo M2 proposto com o modelo de [Agustina, Lee e Piplani \(2014\)](#). Observamos que o Modelo M2 apresentou um melhor desempenho. Possivelmente, essa vantagem está relacionada às variáveis de dois índices utilizadas para a modelagem das decisões de roteamento diferentemente de [Agustina, Lee e Piplani \(2014\)](#) que utilizam variáveis de três índices.

Os experimentos computacionais realizados mostram a relevância da integração dos problemas de programação de *cross-dock* e de roteamento de veículos na definição de um plano de distribuição do sistema. Principalmente para as instâncias com 15 clientes, as vantagens em resolver os dois problemas integrados ficam mais evidentes e as soluções obtidas apresentam uma melhor qualidade comparadas às soluções das outras estratégias. A estratégia RCD, em que o problema de roteamento é resolvido primeiro, foi a que teve desempenho mais próximo à estratégia EI. Em contrapartida, com a RCD, verificamos que é possível obter rotas não viáveis para o sistema, ou seja, rotas que não são possíveis ser completadas antes do tempo limite. Além disso, ao resolvermos os problemas separadamente, não garantimos um plano ótimo de distribuição. No caso das instâncias com 25 clientes, as melhores soluções são obtidas com as estratégias EI e RCD que apresentam desempenhos muito próximos, com uma pequena vantagem da EI. Mesmo assim, considerando os *gaps* das soluções obtidas com a EI, existem possibilidades de maiores ganhos, principalmente no Cenário 2, em que os *gaps* são maiores (18% em média). Na resolução das instâncias maiores com 40 clientes, conforme esperado, tivemos

maior dificuldade em resolver com a estratégia integrada, sendo que, para algumas instâncias, não foi possível obter solução viável em 2 horas e grande parte das instâncias, principalmente no Cenário 2, apresentaram *gaps* altos. Neste caso, as melhores soluções são obtidas com a estratégia RCD.

Com o que foi apresentado neste capítulo, podemos concluir que, de fato, a integração entre os problemas de programação *cross-dock* e de roteamento de veículos permite obter um plano de distribuição mais eficiente, com menor custo total e um menor tempo de violação das janelas de tempos de entrega dos clientes. Este último, reflete diretamente numa maior satisfação dos clientes. Porém, verificamos que, à medida que a dimensão do problema aumenta, fica mais difícil resolvê-lo de forma integrada. Diante disso, para tratar instâncias maiores e obter soluções em menor tempo, desenvolvemos métodos de solução heurísticos para o problema estudado nos próximos capítulos. Em especial, no capítulo seguinte, apresentamos uma heurística construtiva.

HEURÍSTICA CONSTRUTIVA

O estudo das estratégias integrada e hierárquicas para a resolução dos problemas de programação do *cross-dock* e de roteamento das cargas de entrega mostrou que, de fato, a resolução integrada dos problemas resulta em uma programação do sistema de distribuição com menores custos totais e menor número de violações nos prazos das entregas. No entanto, como esperado, a estratégia integrada demanda mais tempo computacional para resolução que as estratégias hierárquicas, tornando-se inviável para problemas de grande dimensão.

De acordo com os experimentos computacionais realizados (ver Capítulo 3), a estratégia hierárquica que leva a resultados mais próximos dos obtidos pela estratégia integrada, em termos de custo total, é a estratégia em que primeiro é resolvido o problema de roteamento de veículos, ou seja, são definidas as rotas de entrega, e, em seguida, é planejada a programação do recebimento e do carregamento das cargas. Diante disso e da necessidade de um método de resolução mais rápido para tratar os problemas de maior dimensão, propomos uma heurística construtiva baseada na esta estratégia hierárquica para obter soluções para o problema. A ideia central da heurística proposta é determinar as rotas para as entregas e, paralelamente, as decisões de designação e de ordenação para o processamento das cargas no *cross-dock*.

4.1 Algoritmo da heurística construtiva

De forma geral, a heurística consiste em construir as rotas de entrega uma a uma. Para a construção de cada uma das rotas são considerados dois critérios: (i) critério de prioridade dos clientes; e (ii) critério de compatibilidade entre clientes. O critério de prioridade determina o cliente pelo qual uma rota tem início, enquanto o critério de compatibilidade define qual o próximo cliente a ser adicionado à rota, levando em consideração o último cliente visitado na rota em construção. Assim, à medida em que as rotas são construídas, alocamos as cargas de entrada necessárias para a consolidação da carga de saída associada às rotas e designamos a doca de saída onde cada carga consolidada é carregada. A ideia geral da heurística é resumida no

Algoritmo 1. Os passos da heurística bem como os critérios de prioridade e de compatibilidade utilizados são detalhados nas próximas seções.

Algoritmo 1 – Heurística construtiva

Entrada: Dados do problema relacionados ao processamento e à consolidação das cargas no *cross-dock* e ao conjunto de clientes $\mathcal{C}$ (demanda, tempo de serviço, tempo de deslocamento e janelas de tempo).

Saída: Solução: designação e ordenação para o processamento das cargas de entrada e de saída, consolidação e rotas de entrega.

- 1: Inicialize o índice de rotas ($r = 0$).
 - 2: Ordene os clientes ($\mathcal{C}$) de acordo com o **Critério de Prioridade**.
 - 3: **enquanto** $\mathcal{C} \neq \emptyset$ **faça**
 - 4: Escolha o cliente $c \in \mathcal{C}$ com maior prioridade para iniciar uma nova rota ($r = r + 1$).
 - 5: Faça $\mathcal{C} = \mathcal{C} - \{c\}$.
 - 6: Defina a **Escolha da doca de saída** para a rota r .
 - 7: Atualize o **Sequenciamento das Cargas** para atender o cliente c .
 - 8: Faça $\mathcal{C}_c = \mathcal{C}$.
 - 9: **enquanto** $\mathcal{C}_c \neq \emptyset$ **faça**
 - 10: Ordene os clientes de $\mathcal{C}_c$ de acordo com o **Critério de Compatibilidade** com o cliente c ($Lista_c$).
 - 11: Escolha o cliente $cl \in Lista_c$ mais compatível.
 - 12: **se** **Inserção** (cl na rota r) = Verdadeiro **então**
 - 13: Inclua cl na rota r .
 - 14: Atualize o **Sequenciamento das Cargas** para atender o cliente cl .
 - 15: Faça $\mathcal{C} = \mathcal{C} - \{cl\}$ e $c = cl$.
 - 16: Faça $\mathcal{C}_c = \mathcal{C}_c - \{cl\}$.
-

4.2 Critério de Prioridade

O critério de prioridade de cada cliente é calculado levando em consideração o tempo mínimo necessário para que sua entrega chegue até o seu destino e o término de sua janela de tempo. Logo, primeiramente, calculamos o limitante inferior (lb_i) para o instante de chegada da entrega até o cliente i (t_i), caso ele seja o primeiro a ser atendido em uma rota, ou seja, o tempo mínimo necessário para descarregar os produtos demandados pelo cliente, enviá-los à doca de saída, carregá-los no veículo e, depois, transportá-los até seu destino. O limitante (lb_i) é dado por:

$$lb_i = UT_{\alpha_i^*} + \tau_i^* + Q_i LT + TT_{0i}. \quad (4.1)$$

em que:

- $UT_{\alpha_i^*}$: é o máximo entre os instantes de término de processamento das cargas demandadas pelo cliente i , supondo que elas são processadas o mais cedo possível ($\alpha_i^* = \underset{l \in \bar{L}_i}{\operatorname{argmax}} \{UT_l\}$) em que $\bar{L}_i$ é o conjunto de cargas demandadas pelo cliente i ;

- τ_i^* : é o tempo mínimo para transferir os produtos demandados pelo cliente i da carga de entrada α_i^* para a doca de saída onde a carga será carregada;
- Q_iLT : é o tempo de carregamento da carga do cliente i ;
- TT_{0i} : é o tempo de trajeto do veículo partido do *cross-dock* para o cliente i .

Para o cálculo de $UT_{\alpha_i^*}$, uma ordem de processamento das cargas de entrada é simulada alocando-as na doca que primeiro estiver livre e em ordem decrescente dos seus tempos de processamento.

Depois de obtido lb_i , calculamos a prioridade do cliente i , que é dada por:

$$pr_i = B_i - lb_i \quad (4.2)$$

em que B_i é o fechamento da janela de tempo para atender o cliente i .

Dessa forma, quanto maior o resultado dessa diferença, menor a prioridade do cliente i , pois seu pedido pode ser despachado depois de outros. Portanto, na *linha 2*, a ordem de prioridade é dada de acordo com ordem crescente de pr_i . Esse critério pode ser estático (Pr1) quando é calculado apenas no início do algoritmo, ou dinâmico (Pr2), neste caso, é sempre atualizado antes que uma nova rota seja iniciada levando em consideração a ordem de processamento das cargas de entrada que já tenham sido alocadas, ou seja, um novo passo de atualização do critério de prioridade é incluído no Algoritmo 1 logo depois da *linha 16*.

4.3 Critério de Compatibilidade

O critério de compatibilidade guia a tomada de decisões relacionadas à consolidação das cargas de saída e à ordem das visitas. A ideia de compatibilidade utilizada é procurar por clientes cuja inserção no final de uma determinada rota evite violações das janelas de tempo. Assim, definimos a seguir dois critérios de compatibilidade, um estático e outro dinâmico.

O critério de compatibilidade estático (*Comp1*) para cada par de clientes i e j é calculado da seguinte forma:

$$comp_{ij} = \begin{cases} \infty, & \text{se } A_i + Q_iST + TT_{ij} > B_j, \\ TT_{ij} + T_j^v, & \text{caso contrário;} \end{cases}$$

em que: B_j é o limite superior da janela de tempo de atendimento do cliente j ; e T_j^v representa o pior caso para o tempo de violação da janela de tempo do cliente j , supondo que ele seja adicionado a rota na sequência do cliente i . Logo, o valor de T_j^v é dado por:

$$T_j^v = \max \{Antes, Depois\}.$$

em que:

- se a carga chegar no cliente j antes do início de sua janela de tempo de entrega, a antecipação é dada por $Antes = \max \{0, A_j - (\frac{A_i+B_i}{2} + Q_iST + TT_{ij})\}$, ou seja, a diferença (positiva) entre o limite inferior da janela de tempo do cliente j e o mais cedo que é possível chegar em j depois de deixar o cliente i , supondo que o cliente i foi visitado no instante igual ao meio de sua janela de tempo ($\frac{A_i+B_i}{2}$), contabilizando o tempo de descarregamento no cliente i (Q_iST) e o tempo de trajeto do cliente i para o cliente j (TT_{ij});
- se a carga chegar no cliente j depois do término de sua janela de tempo de entrega, o atraso é dado, analogamente ao item anterior, por $Depois = \max \{0, (\frac{A_i+B_i}{2} + Q_iST + TT_{ij}) - B_j\}$.

No caso dinâmico (Comp2), a compatibilidade entre os clientes i e j é calculada a partir do instante de chegada em i de acordo com a construção da solução e não a partir do instante médio da janela de tempo de i como acontece no caso estático. Assim, $comp_{ij}$ segue a mesma regra de cálculo definida para o caso estático trocando apenas $\frac{A_i+B_i}{2}$ pelo instante de chegada no cliente i (AT_i) que é definido à medida em que a solução é construída.

Quando $comp_{ij}$ é igual a infinito, consideramos que o cliente j é incompatível com o cliente i pois, se as entregas dos clientes i e j ocorrerem de forma consecutiva numa mesma rota, temos certamente custos com atraso na entrega para o cliente j , mesmo quando o veículo chegar ao cliente i no início de sua janela de tempo. Caso contrário, sua compatibilidade ($comp_{ij}$) é dada pela soma do tempo de trajeto entre os clientes i e j com o tempo de violação da janela de tempo do cliente j ao simular o pior caso considerando o instante da visita no cliente i . Assim, o cliente mais compatível com o cliente i é o cliente $j \in \mathcal{C}$ tal que $comp_{ij}$ é finito e tem o menor valor, já que as parciais de $comp_{ij}$ aparecem na função objetivo que é de minimização. E, no caso do menos compatível, é tal que $comp_{ij}$ é finito e tem o maior valor. Dessa forma, na linha 11, denotamos por $Lista_c$ a lista de todos os clientes em ordem crescente do critério de compatibilidade em relação a um dado cliente.

4.4 Escolha da doca de saída

Na linha 6 do Algoritmo 1, quando é escolhida a doca de saída para o carregamento da carga de saída associada à rota r , escolhemos sempre a doca de saída que primeiro estiver livre, i.e., a doca com o menor tempo de liberação considerando os carregamentos das cargas de saída associadas às rotas anteriormente definidas. Assim, estamos também fixando a ordenação dos carregamentos.

4.5 Sequenciamento de cargas

Quando uma nova rota é iniciada ou um novo cliente é inserido numa rota já existente (linhas 7 e 14 do Algoritmo 1, respectivamente), é preciso verificar a necessidade de sequenciar

novas cargas de entrada nas docas de entrada. Além disso, também é necessário complementar as transferências de produtos entre docas de entrada e de saída. Na heurística desenvolvida, cada nova carga de entrada é designada a doca de entrada que tiver o tempo mais baixo de processamento das cargas a ela atribuídas, ou seja, a que estiver livre primeiro. Quando houver mais de uma carga de entrada a ser designada, a designação é feita em ordem decrescente de seus tempos de processamento. Dessa forma, buscamos que a liberação da carga de saída de uma rota seja o mais rápido possível.

4.6 Inserção de cliente

Para verificar a possibilidade de inclusão de um cliente em uma dada rota (*linhas* 12-15 do Algoritmo 1), verificamos três condições: (i) capacidade disponível no veículo; (ii) instante de entrega no primeiro da rota; e (iii) instante de retorno do veículo para o *cross-dock*.

A primeira condição impõe que só podemos incluir o cliente j na rota se a capacidade do veículo não é violada, ou seja, se há capacidade disponível no veículo para o carregamento da carga do cliente j . Quando um cliente é adicionado a uma rota, os produtos por ele demandados devem ser transferidos à doca de saída, logo devem ser descarregados e direcionados para esta doca. Podemos ter, portanto, um aumento do tempo de despacho da carga. Assim, pela segunda condição, o cliente j só pode ser adicionado a rota se o instante da entrega para o primeiro cliente dessa rota não ultrapassa o término de sua janela de tempo. A terceira condição assegura que se o cliente j for adicionado a rota, o instante de retorno do veículo ao *cross-dock* não ultrapassa o tempo limite para o retorno dos veículos ao *cross-dock*.

4.7 Obtenção do valor da função objetivo

Ao longo da construção da solução, o tempo total de viagem e o *makespan* que fazem parte da função objetivo são facilmente obtidos. No entanto, o cálculo das violações das janelas de tempo não é trivial. Portanto, para obter o valor da função objetivo com o tempo total de antecipação e de atraso mínimo, resolvemos, com o auxílio do *solver*, o Modelo M2 com as decisões de rotas (sequências das visitas), de designação e de ordenação das cargas conforme definido pela heurística construtiva. Com essas decisões fixadas, o modelo se torna um problema de programação linear de fácil resolução.

4.8 Experimentos computacionais

Os experimentos computacionais foram realizados usando o *solver* Gurobi (versão 8.1) na sua configuração padrão em um computador Intel Core i7-2600 2,40Gz com 15,6 GB (Linux 18.04 LTS). A heurística foi implementada em linguagem C++. Para os testes, utilizamos os

mesmos três conjuntos de instâncias com 15, 25 e 40 clientes dos experimentos computacionais apresentados no Capítulo 3.

Combinado os critérios de prioridade (Pr1 e Pr2) e de compatibilidade (Comp1 e Comp2), temos quatro versões para a heurística construtiva. Os testes foram realizados para avaliar a qualidade das soluções obtidas. Para isso, comparamos as soluções das heurísticas com as soluções obtidas pelo *solver* com tempo limite de 1800 segundos. Apenas para as instâncias com 40 clientes, em que não foi possível obter soluções em 1800 segundos para 37 instâncias das 72 instâncias (6 instâncias do Cenário 1 e 31 instâncias do Cenário 2), limitamos o tempo a 2 horas. Mesmo assim, não foram obtidas soluções para 7 instâncias do Cenário 2. A seguir, uma análise de qualidade das soluções obtidas pelas quatro versões da heurística construtiva é resumida nas Tabelas 14 e 15. Ao final, é discutido o desempenho das heurísticas em relação ao tempo de execução.

4.8.1 Resultados computacionais - Cenário 1

Na Tabela 14, são reportados: o número de soluções infactíveis obtidas (SI) e a média do tempo extra utilizado (TE). Além disso, também são apontados os valores mínimo (D^{min}) e máximo (D^{max}), a média ($\bar{D}$) e o desvio padrão (D^σ) das diferenças percentuais (D) entre as soluções obtidas pelo *solver* e pela heurística (factíveis e infactíveis) para cada conjunto de instância de acordo com o número de clientes. Em negrito, estão as médias desses valores para cada versão da heurística. O cálculo das diferenças percentuais é definido da seguinte forma:

$$D = \frac{SOL^{M2} - SOL^H}{SOL^{M2}} \times 100,$$

onde SOL^{M2} é a solução obtida pelo *solver* ao resolver o Modelo M2 e SOL^H é solução obtida pela heurística. Os resultados apresentados aqui estão focados na avaliação da heurística, porém mais detalhes sobre as soluções obtidas pelo *solver* e sobre as soluções heurísticas são apresentados nos Apêndices B e C, respectivamente.

Para o Cenário 1, a maioria das soluções obtidas pelas heurísticas são infactíveis, pois o tempo necessário para completar todas as entregas é maior que o tempo limite (T^{max}). Assim, foi necessário adicionar um tempo extra (TE) para completá-las. Conforme pode ser observado na Tabela 14, o número de soluções infactíveis aumenta para instâncias com maior número de clientes (40 clientes). Possivelmente, isso está relacionado ao fato de que a heurística prioriza as decisões de roteamento e, no Cenário 1, elas são menos relevantes, pois temos menos possibilidades de consolidações já que as demandas dos clientes são maiores, entre 4 e 16 paletes. Em geral, verificamos que essas soluções poderiam ser factibilizadas com uma nova designação e ordenação das cargas de saída, ou seja, utilizando uma fase de melhoria da solução. Todas as soluções heurísticas apresentaram valores da função objetivo piores que as soluções obtidas pelo *solver*, variando em média de 33,7% a 41,4% pior. Vale destacar que no melhor caso as heurísticas obtiveram soluções muito próximas das soluções do *solver*, variando de 3,7% a 11,0% pior.

Tabela 14 – Performance das versões da heurística em relação ao *solver* - Cenário 1.

		SI	TE	D^{min}	D^{max}	$\bar{D}$	D^σ
H(Pr1,Comp1)	15 clientes	16	9,1	-64,7	-3,6	-21,6	15,9
	25 clientes	29	31,7	-149,4	-9,1	-35,8	27,4
	40 clientes	36	48,7	-147,3	-9,5	-43,6	30,6
		27	29,8	-120,5	-7,4	-33,7	24,6
H(Pr1,Comp2)	15 clientes	16	9,1	-64,7	-3,6	-21,6	15,8
	25 clientes	29	31,9	-149,4	-9,1	-35,8	27,5
	40 clientes	36	50,1	-172,2	-9,5	-45,2	32,8
		27	32,4	-128,8	-7,4	-34,2	25,4
H(Pr2,Comp1)	15 clientes	20	11,7	-111,5	-3,7	-23,2	21,6
	25 clientes	31	38,1	-165,2	-11,0	-41,7	30,8
	40 clientes	35	56,3	-154,8	-5,8	-52,8	41,7
		29	35,4	-143,8	-6,8	-39,2	31,4
H(Pr2,Comp2)	15 clientes	20	11,7	-111,5	-3,7	-29,5	21,5
	25 clientes	31	39,0	-165,2	-9,9	-41,9	31,1
	40 clientes	36	55,9	-154,8	-5,8	-52,8	41,7
		29	35,5	-143,8	-6,5	-41,4	31,4

SI: número de soluções infactíveis TE: tempo extra $\bar{D}$: diferença percentual média D^{min} : diferença percentual mínima D^{max} : diferença percentual máxima D^σ : desvio padrão da diferença percentual

Fonte: Elaborada pelo autor.

De forma geral, as versões da heurística que utilizam o critério de prioridade Pr1 apresentaram um melhor desempenho, obtendo um maior número de soluções factíveis e com média e desvio padrão das diferenças percentuais melhores. Com relação ao critério de compatibilidade as diferenças são menos significativas, no entanto, a H(Pr1,Comp1) apresenta resultados melhores que as demais. Note que, mesmo com critérios de compatibilidade diferentes, as duas versões apresentam uma performance semelhante reforçando mais uma vez que as decisões de roteamento neste cenário não têm grande impacto, já que esses critérios estão diretamente relacionados à construção das rotas. Dessa forma, dado que o critério de prioridade está mais relacionado às decisões de designação e de ordenação das cargas de saída, podemos concluir que essas decisões têm mais impacto sobre a solução do problema no Cenário 1.

4.8.2 Resultados computacionais - Cenário 2

No Cenário 2, foram obtidas soluções factíveis para quase todas as instâncias, apenas para o conjunto de instâncias com 40 clientes as versões H(Pr1,Comp1) e H(Pr1,Comp2) geraram soluções infactíveis para duas instâncias com tempo extra de 21,4 e 16,7 e a versão H(Pr2,Comp1) gerou uma solução infactível para uma instância com tempo extra de 2,61. De forma análoga ao Cenário 1, os resultados para o Cenário 2 estão resumidos na Tabela 15. Como praticamente todas as soluções são factíveis, não apresentamos aqui as colunas relacionadas ao número de

soluções infactíveis e ao tempo extra.

Tabela 15 – Performance das versões da heurística em relação ao *solver* - Cenário 2.

		D^{min}	D^{max}	$\bar{D}$	D^σ
H(Pr1,Comp1)	15 clientes	-44,7	-9,4	-24,2	8,9
	25 clientes	-51,7	-8,4	-27,2	11,5
	40 clientes	-60,8	5,9	-22,2	16,9
		-52,4	-4,0	-24,5	12,5
H(Pr1,Comp2)	15 clientes	-47,5	-9,0	-23,1	9,7
	25 clientes	-51,7	-7,6	-24,4	10,6
	40 clientes	-56,3	5,7	-20,2	17,1
		-51,8	-3,6	-22,5	12,5
H(Pr2,Comp1)	15 clientes	-42,0	-8,2	-24,3	8,0
	25 clientes	-86,0	-5,7	-27,8	13,6
	40 clientes	-72,4	4,3	-23,1	16,6
		-66,8	-3,2	-25,1	12,7
H(Pr2,Comp2)	15 clientes	-42,0	-9,7	-23,7	8,8
	25 clientes	-89,1	-3,1	-26,3	14,0
	40 clientes	-70,8	5,7	-20,8	18,2
		-67,3	-2,4	-23,6	13,7

$\bar{D}$: diferença percentual média D^{min} : diferença percentual mínima

D^{max} : diferença percentual máxima D^σ : desvio padrão da diferença percentual

Fonte: Elaborada pelo autor.

De forma geral, no Cenário 2, as diferenças percentuais são menores e indicam uma maior proximidade entre as soluções heurísticas e as soluções obtidas pelo *solver*. Note que, no conjunto de instâncias com 40 clientes, as quatro versões forneceram soluções melhores (até 5,9%) que as soluções obtidas pelo *solver* para duas instâncias. É possível observar que, embora o tempo limite para o *solver* tenha sido maior para as instâncias com 40 clientes, as médias das diferenças percentuais são menores se comparadas às médias das diferenças percentuais no caso das instâncias com 25 clientes. Conforme esperado, isso sinaliza que a solução heurística é mais vantajosa à medida que o número de clientes aumenta. Ao compararmos as performances das diferentes versões da heurística, não temos uma performance significativamente melhor, apenas H(Pr1,Comp2) tem uma diferença percentual média um pouco menor que as diferenças percentuais média apresentadas pelas outras.

4.8.3 Tempo de solução das instâncias

Para uma análise do ponto de vista de tempo de solução, na Tabela 16, são reportadas as médias dos tempos de resolução (em segundos) de cada versão da heurística de acordo com o cenário e o número de clientes. As quatro versões da heurística apresentam tempo de resolução praticamente iguais, logo, na tabela, apresentamos apenas os tempos médios. Podemos

observar que obter uma solução utilizando a heurística construtiva proposta é bastante rápido e, principalmente no Cenário 2, as soluções podem estar próximas, com diferença percentual média de 23,9%, ou serem até melhores que algumas soluções fornecidas pelo *solver* como ocorreu para duas instâncias com 40 clientes. Além disso, é importante ressaltar que não foi possível obter soluções para 7 instâncias com 40 clientes no Cenário 2 utilizando o *solver*, enquanto que, com a heurística, são obtidas soluções factíveis para essas 7 instâncias.

Tabela 16 – Tempos de solução: heurística $\times$ *solver* (em segundos).

	Cenário 1		Cenário 2	
	Heurística	Solver	Heurística	Solver
15 clientes	<1	616	<1	1541
25 clientes	<1	1495	<1	1800
40 clientes	10	7200	10	7200

Fonte: Elaborada pelo autor.

4.9 Conclusões

De forma geral, podemos concluir que a heurística construtiva é bastante rápida e apresentou um bom desempenho, principalmente, para obter soluções factíveis para as instâncias do Cenário 2. Em especial, para as instâncias com 40 clientes, foi possível obter em poucos segundos boas soluções se comparadas às soluções fornecidas pelo *solver* no tempo de 2 horas. Além disso, obtivemos soluções factíveis para 7 instâncias para as quais o *solver* não obteve soluções.

No caso do Cenário 1, em que temos menos possibilidades de consolidações de cargas, as decisões de roteamento são menos relevantes, não tivemos um bom desempenho da heurística em relação à qualidade das soluções, pois a maioria delas é infactível. Possivelmente, como já apontado, isso ocorre por conta da heurística priorizar as decisões de roteamento. Uma estratégia para resolver o problema de infactibilidade seria uma redefinição da designação e da ordenação das cargas de saída que pode ser incorporada, por exemplo, através de uma busca local.

Entre as versões testadas, no Cenário 1, as duas nas quais são utilizadas o critério de prioridade Pr1 apresentam desempenho bastantes similares e melhores que as outras, em especial, H(Pr1,Comp1) apresenta uma pequena vantagem sobre H(Pr1,Comp2). Enquanto que, no Cenário 2, tivemos a H(Pr1,Comp2) com melhor desempenho. Além disso, de forma geral, a heurística construtiva tem um tempo computacional baixo e pode ser um ponto de partida na elaboração de outros métodos de resolução mais elaborados. Nesse sentido, no próximo capítulo apresentamos uma meta-heurística cuja solução inicial é dada pela heurística construtiva.

META-HEURÍSTICA: BUSCA EM VIZINHANÇA VARIÁVEL

O método Busca em Vizinhança Variável (*Variable Neighborhood Search* - VNS) é uma meta-heurística proposta na década de noventa por [Mladenovic \(1995\)](#) e [Mladenović e Hansen \(1997\)](#). A ideia principal do método é o uso de diferentes tipos de vizinhanças de forma a ampliar o espaço de busca aumentando as chances de obter soluções mais próximas da solução ótima. De acordo com [Hansen e Mladenović \(2001\)](#), ao longo da busca, muitas variáveis podem assumir seus valores ótimos em uma dada solução incumbente e, possivelmente, eles serão mantidos e usados para obter soluções vizinhas melhores; isto é, diferentes ótimos locais em relação a vizinhanças distintas podem apresentar estruturas semelhantes. Esse é um dos princípios citado em [Hansen e Mladenović \(2001\)](#) sobre o qual o método está baseado. Além desse princípio, os autores também listam outros dois: um ótimo local em uma vizinhança não necessariamente é um ótimo local em outra vizinhança, enquanto um ótimo global é um ótimo local em qualquer vizinhança.

Diferentes abordagens integradas de métodos de soluções que usam a VNS como base podem ser encontradas na literatura. [Li et al. \(2017\)](#) observa que esse tipo de abordagem é bastante comum na literatura de problemas de dimensionamento de lotes e citam alguns trabalhos neste contexto. Em especial, [Chen \(2015\)](#) e [Li et al. \(2017\)](#) combinam *Fix-and-Optimize* e VNS para o problema de dimensionamento de lotes multiníveis, sendo que o primeiro aborda uma versão determinística do problema, enquanto o segundo aborda uma versão estocástica. Recentemente, [Toschi et al. \(2018\)](#) propuseram um método de solução combinando *Fix-and-Optimize* e VNS para o problema de transporte de amostras biomédicas (*biomedical sample transportation problem*) com o qual são obtidas boas soluções para instâncias reais em um tempo bem menor quando comparado à resolução do problema utilizando diretamente o *solver* CPLEX. Este último trabalho, como é uma variação do problema de roteamento no contexto de logística na área da saúde, é mais próximo ao nosso problema. Os autores usam a estratégia

Fix-and-Optimize para efetuar as buscas locais através do próprio modelo de programação inteira mista proposto para o problema, no qual são fixados os valores de algumas variáveis de acordo com as estruturas de vizinhanças definidas e uma restrição de corte também é adicionada impondo a busca por uma solução melhor que a incumbente. Com essa estratégia para as buscas locais, os autores estruturam a VNS em três fases. A primeira é uma fase de intensificação em que uma busca local é executada com uma vizinhança mais restrita. Na segunda, denominada diversificação, as vizinhanças têm suas dimensões aumentadas e a busca local é reiniciada. A terceira fase é definida de forma a evitar que a busca fique presa a um mínimo local. Para isso, executa-se uma busca local permitindo obter uma solução pior que a incumbente respeitando uma dada tolerância. A seguir, são apresentados detalhes sobre a meta-heurística desenvolvida para o problema estudado aqui.

5.1 Algoritmo da meta-heurística

O Algoritmo 2, baseado em Hansen e Mladenović (2001), representa os passos básicos da VNS desenvolvida. Uma vez definidas as vizinhanças, obtida uma solução inicial s e definido o critério de parada que pode ser, por exemplo, o tempo de execução ou número máximo de iterações, inicia-se o processo de busca para refinar a solução incumbente conforme os passos descritos. Observe, na linha 3, a existência de um fator de aleatoriedade que tem como objetivo a diversificação das soluções. Esse passo pode não fazer parte de uma VNS e, neste caso, o método é denominado na literatura mais especificamente por Vizinhança Variável Descendente (*Variable Neighborhood Descent* - VND). Nessa variante, as soluções vizinhas são obtidas de forma determinística. Assim, a partir de uma dada solução viável, a busca é executada de acordo com a vizinhança definida em cada iteração. Caso seja obtida uma solução melhor s'' , executa-se outra busca de acordo com a vizinhança N_1 tendo como ponto de partida s'' ; caso contrário, outra busca com a próxima vizinhança é realizada a partir da solução incumbente atual.

Algoritmo 2 – *Variable Neighborhood Search*

Entrada: Um conjunto de vizinhanças N_k , $k = 1, \dots, k_{max}$, uma solução inicial s e um critério de parada.

Saída: Solução refinada.

- 1: Inicie com $k = 1$.
 - 2: **enquanto** o critério de parada não for atingido **faça**
 - 3: Obtenha aleatoriamente s' vizinho de s ; isto é, $s' \in N_k(s)$.
 - 4: Aplique um método de busca local com a solução inicial s' e obtenha um ótimo local s'' .
 - 5: **se** s'' é melhor que s' **então**
 - 6: Faça $s = s''$ e $k = 1$.
 - 7: **senão**
 - 8: Faça $k = k + 1$.
-

O método de solução proposto para o problema estudado neste trabalho é uma combinação entre *Fix-and-Optimize* e VND, o qual será denominado por *Fix-and-Optimize VND*

(FO-VND). O FO-VND desenvolvido consiste em sucessivas buscas locais por uma solução melhor que a solução incumbente usando diferentes vizinhanças que são definidas a seguir. Além disso, a solução inicial utilizada é obtida através da heurística construtiva $H(\text{Pr1}, \text{Comp2})$ que, de acordo com os testes computacionais, tem o melhor desempenho. No caso de soluções infactíveis que apresentam um tempo extra para que todas as entregas sejam realizadas e os veículos retornem ao *cross-dock*, esse tempo extra é verificado entre uma busca local e outra de forma que a variável que o define seja fixada em zero no primeiro momento em que ela é nula, ou seja, uma solução factível seja obtida. As soluções são comparadas pelo valor da função objetivo e o critério de parada é o tempo de execução ou se a busca local com a última vizinhança não obtiver uma solução melhor.

Através da estratégia *Fix-and-Optimize*, executamos cada busca local resolvendo o modelo original do problema com algumas decisões fixas de acordo com uma estrutura da vizinhança e com a inclusão de uma restrição de corte que impõe a busca por uma solução de melhor qualidade. Conforme já apontado na descrição do problema, ele é composto por decisões que são bem definidas nas suas diferentes partes, porém elas são fortemente interdependentes. Com base nisso, as vizinhanças foram definidas considerando os tipos de decisões e determinadas partições do problema. Além disso, as vizinhanças foram definidas levando em consideração também a dimensão do subproblema cuja solução deve resultar em uma busca local rápida. A seguir, definimos o modelo de programação inteira mista que é resolvido em cada busca local e as estruturas de vizinhanças utilizadas.

5.2 Modelo para as buscas locais

Cada busca local é executada resolvendo um modelo de programação inteira mista com os valores de algumas variáveis binárias fixos e outros livres de acordo com a estrutura da vizinhança considerada. O tempo limite para cada busca local é 20 segundos. Esse tempo foi definido com base em testes preliminares. O modelo é obtido a partir do Modelo M2 com algumas modificações e com inclusão de uma restrição de corte. O Modelo M2 assim como os parâmetros e as variáveis para sua descrição são apresentados no Capítulo 3. Dada uma vizinhança N_k e uma solução incumbente $s' = (z^{in'}, w^{in'}, z^{out'}, w^{out'}, x', v', et', lt', dt^{max'})$, a busca local, neste caso, consiste em resolver o **Modelo M^{BL}** apresentado a seguir.

$$\min f(x, et, lt, dt^{max}) + 2t^{ex} \quad (5.1)$$

Sujeito a:

$$(3.2) - (3.24)$$

$$t_{N+1} \leq T^{max} + t^{ex} \quad (5.2)$$

$$f(x, et, lt, dt^{max}) \leq f(x', et', lt', dt^{max'}) - \varepsilon \quad (5.3)$$

$$t^{ex} \geq 0 \quad (5.4)$$

$$z^{in} = z^{in'}, w^{in} = w^{in'}, z^{out} = z^{out'}, w^{out} = w^{out'}, x = x', v = v' \quad z^{in}, w^{in}, z^{out}, w^{out}, x, v \notin N_k \quad (5.5)$$

$$z^{in}, w^{in}, z^{out}, w^{out}, x, v \in \{0, 1\} \quad z^{in}, w^{in}, z^{out}, w^{out}, x, v \in N_k \quad (5.6)$$

Como as soluções obtidas pela heurística construtiva H(Pr1,Comp2) não são todas factíveis por apresentarem um tempo extra para completar as entregas e os veículos retornarem ao *cross-dock*, definimos uma variável contínua t^{ex} para representar esse tempo que é penalizado na função objetivo. Essa variável aparece na restrição (5.2) permitindo com que um veículo chegue ao *cross-dock* depois do tempo limite T^{max} . Mas, no primeiro instante em que uma solução é obtida com t^{ex} nula, o valor de t^{ex} é fixado em zero para as buscas posteriores. As restrições (3.2)-(3.24) são as restrições do modelo original. A restrição (5.3) é a restrição de corte que impõe a busca por uma solução com valor da função objetivo menor ou igual ao valor da função objetivo da solução incumbente menos uma dada tolerância ($\varepsilon = 10^{-3}$). As restrições (5.4)-(5.6) definem o domínio das variáveis. Em especial, com a vizinhança N_k considerada, as restrições (5.6) determinam o domínio das variáveis livres e as restrições (5.5) fixam os valores de outras de acordo com a solução incumbente para que não sejam alteradas durante a busca local.

5.2.1 Inequações válidas

Da forma como o problema estudado foi modelado, identificamos que as variáveis y_{li} , w_{lm}^{in} e w_{ij}^{out} , dependendo dos valores definidos para outras variáveis, podem assumir quaisquer valores (0 ou 1) em uma determinada solução sem alterar o valor da função objetivo e as decisões que realmente definem a programação do *cross-dock*. Ao efetuar as buscas locais, é necessário ter uma atenção especial com esses valores que são desnecessários na definição da solução e, caso sejam fixados, a busca local pode ficar restrita, eliminando algumas mudanças. Uma maneira de evitar que essas variáveis assumam esses valores é com a inclusão de inequações válidas, definidas a seguir. Como sabemos, inequações válidas, quando adicionadas ao modelo, não eliminam nenhuma solução viável para o problema.

No caso da variável y_{li} , mesmo se não houver transferências de produtos da carga de entrada l para a carga de saída i , a variável y_{li} pode assumir o valor 1 sem alterar uma dada solução. Por exemplo, se o instante de término do processamento da carga de entrada l (ut_l) for menor que o instante de liberação da carga de saída i (rt_i), y_{li} pode ser igual a 1 sem nenhuma consequência para a solução, já que não interfere no tempo de disponibilidade da carga de saída i . Além disso, caso a carga de saída i não exista, ou seja, nenhuma rota inicia com a entrega do cliente i , y_{li} também pode assumir o valor 1 sem afetar a solução. Isso acontece porque a variável y_{li} não é penalizada diretamente na função objetivo e as restrições (3.13) e (3.14), definidas no

Capítulo 3, não são violadas. Neste último caso, para eliminar esses valores em uma solução, incluímos as inequações válidas (5.7) no Modelo M^{BL} .

$$y_{li} \leq x_{0i} \quad \forall l \in \mathcal{L}; i \in \mathcal{N} \quad (5.7)$$

As inequações (5.7) apenas permitem que haja transferências para as cargas de saída que realmente existem, já que as rotas (cargas de saída) são rotuladas de acordo com o índice do primeiro cliente visitado, ou seja, se a variável x_{0i} é igual a 1. Assim, se x_{0i} é nula, y_{li} também é nula para todo l .

No caso das variáveis w_{lm}^{in} , que determinam a ordenação das cargas de entrada designadas para uma mesma doca, se as cargas l e m são designadas para docas diferentes e o tempo de término do processamento da carga l é menor que o tempo de término da carga de entrada m , então w_{lm}^{in} pode assumir 1 sem afetar a solução. Novamente, isso acontece porque essas variáveis não são penalizadas na função objetivo e não violam as restrições (3.3) e (3.5), definidas no Capítulo 3. A justificativa é análoga para as variáveis w_{ij}^{out} que determinam a ordenação das cargas de saída. Para impedir que esses valores desnecessários apareçam nas soluções, incluímos as inequações válidas (5.8) e (5.9) no Modelo M^{BL} .

$$w_{lm}^{in} + w_{ml}^{in} \leq 2 - z_{lf}^{in} - z_{lf'}^{in} \quad \forall l, m \in \mathcal{L} (l < m); f, f' \in \mathcal{F} (f < f') \quad (5.8)$$

$$w_{ij}^{out} + w_{ji}^{out} \leq 2 - z_{ih}^{in} - z_{jh'}^{in} \quad \forall i, j \in \mathcal{N} (i < j); h, h' \in \mathcal{H} (h < h') \quad (5.9)$$

As inequações (5.8) e (5.9), respectivamente, anulam as variáveis w_{lm}^{in} e w_{ij}^{out} quando as cargas de entrada l e m e as cargas de saída i e j são designadas para docas diferentes.

Testes preliminares mostraram que essas inequações válidas não melhoram a relaxação linear e nem o desempenho do *solver* ao resolver as instâncias com o Modelo M2. Por isso, elas são consideradas apenas no Modelo M^{BL} para as buscas locais em que contribuem para um melhor desempenho, isso ocorre por contribuírem para a fixação de variáveis para este modelo. Na sequência, apresentamos as vizinhanças utilizadas.

5.3 Vizinhanças utilizadas

Para cada vizinhança é resolvido o problema definido pelo Modelo M^{BL} descrito na Seção 5.2 com parte das variáveis fixadas. Vale destacar que se busca a melhor solução vizinha restrita ao tempo limite de 20 segundos. A seguir, explicamos cada uma das vizinhanças.

5.3.1 Vizinhança N_1 - designação e ordenação na saída

De acordo com a vizinhança N_1 , as soluções são consideradas vizinhas se elas têm as mesmas decisões para o processamento das cargas de entrada, para as consolidações e para a sequência das entregas (rotas). Neste caso, para realizar a busca local, baseando-se na estratégia

Fix-and-Optimize, as variáveis relacionadas a estas decisões são fixadas, ou seja, as variáveis z_{lf}^{in} , w_{lm}^{in} , v_{ij} e x_{ij} são fixadas no modelo nos valores da solução incumbente atual. Portanto, liberam-se as decisões em relação à designação e à ordenação das cargas de saída consolidadas através variáveis z_{ih}^{out} e w_{ij}^{out} sobre as quais o modelo é otimizado em busca de uma solução de melhor qualidade. Em resumo, a ideia desta vizinhança é permitir que as cargas de saída tenham sua designação e ordem para o carregamento alteradas, como ilustra o exemplo da Figura 15.

Figura 15 – Exemplo de soluções vizinhas de acordo com a vizinhança N_1 .

Fonte: Elaborada pelo autor.

Na Figura 15, são ilustradas duas soluções vizinhas de acordo com a vizinhança N_1 . As setas vermelhas indicam as alterações realizadas para obter a solução vizinha de acordo com N_1 . Note que, é possível alterar a posição e a ordem em que as cargas de saída são processadas. É importante ressaltar que o número de cargas e as rotas de cada carga permanecem as mesmas em soluções vizinhas de acordo com N_1 .

5.3.2 Vizinhança N_2 - designação e ordenação na entrada

Na vizinhança N_2 , são considerados os mesmos movimentos definidos na vizinhança N_1 , porém no contexto do processamento das cargas de entrada; ou seja, as decisões de designação e ordenação das cargas de entrada estão livres, enquanto que as outras decisões do problema não se alteram.

Na Figura 16, estão ilustradas duas soluções vizinhas de acordo com a vizinhança N_2 , as setas em vermelho indicam as alterações realizadas para obter a solução vizinha. De forma análoga à vizinhança N_1 , a doca em que cada carga de entrada é processada e a ordem podem ser distintas.

5.3.3 Vizinhança N_3 - ordenação e roteamento em uma doca de saída

A vizinhança N_3 permite movimentos relacionados às decisões de ordenação e de roteamento de cargas de saída que são designadas para uma mesma doca. Assim, de uma solução vizinha para outra, é possível ter diferenças entre o número de cargas de saída, na ordem de

Figura 16 – Exemplo de soluções vizinhas de acordo com a vizinhança N_2 .

Fonte: Elaborada pelo autor.

processamento, na consolidação e na sequência das entregas (rotas) apenas na doca considerada. As decisões de designação e de ordenação das cargas de entrada e as decisões associadas às outras cargas de saída que não estão na doca considerada permanecem inalteradas. Portanto, a busca local com a vizinhança N_3 é a resolução do problema original restrito aos clientes servidos em uma mesma doca com as decisões para o processamento das cargas de entrada fixas. Note que é possível ter soluções vizinhas com diferentes números de cargas de saída processadas na doca considerada. Neste caso, a busca local com a vizinhança N_3 é executada uma vez para cada doca.

Figura 17 – Exemplo de soluções vizinhas de acordo com a vizinhança N_3 .

Fonte: Elaborada pelo autor.

Na Figura 17, são ilustradas duas soluções vizinhas considerando a doca em que as cargas A, D e G são processadas de acordo com a vizinhança N_3 . Na parte debaixo da figura, são ilustradas as rotas de cada uma das cargas. Observe que, além da diferença na ordem do

processamento das cargas de uma solução para outra, as rotas associadas às cargas de saída também são diferentes. Além disso, se tivéssemos diferentes números de cargas de saída na doca considerada de uma solução para outra, mas com os mesmos clientes servidos por ela, as soluções também seriam vizinhas de acordo com a vizinhança N_3 .

5.3.4 Vizinhança N_4 - designação, ordenação e roteamento em duas docas de saída

A vizinhança N_4 é similar à vizinhança N_3 , porém duas docas são consideradas. Na vizinhança N_4 , os movimentos da vizinhança N_3 são estendidos para todas as cargas processadas em um par de docas de saída. Além disso, também são permitidos movimentos relacionados à designação dessas cargas, ou seja, as docas em que as cargas são processadas podem ser diferentes. Nesta vizinhança, o número de cargas de saída pode também ser diferente como ilustrado pelas soluções vizinhas na Figura 18. Além disso, observe que há diferenças entre as decisões nas consolidações e nas rotas das cargas de saída processadas nas duas docas destacadas num retângulo em linha pontilhada vermelha.

Figura 18 – Exemplo de soluções vizinhas de acordo com a vizinhança N_4 .

Fonte: Elaborada pelo autor.

Neste caso, a busca local é executada para todo par de docas respeitando a seguinte ordem. Primeiro, são avaliados os pares de docas adjacentes. Na sequência, os pares que estão separados por uma doca, depois, os pares que estão separados por duas docas e assim por diante até a busca local ser executada para todos os pares de docas. Por exemplo, suponha que o número de docas seja igual a quatro docas D1, D2, D3 e D4 dispostas nessa ordem. Assim, a busca

local para a vizinhança N_4 seria executada seis vezes considerando os pares de docas: (D1,D2), (D3,D4), (D2,D3), (D1,D3), (D2,D4) e (D1,D4), respectivamente.

5.3.5 Vizinhança N_5 - Designação e ordenação de saída e roteamento para cargas em diferentes docas

A vizinhança N_5 é definida de forma a considerar as decisões de roteamento para cargas processadas em docas diferentes. Assim, uma carga de saída de cada doca é escolhida e, para essas cargas escolhidas, as decisões de consolidação e roteamento relacionadas a elas podem ser distintas assim como as docas e as posições em que são processadas. Neste caso, as decisões de ordenação entre as outras cargas de saída (as não escolhidas) também podem ser alteradas, mas as decisões de consolidação e de roteamento para elas permanecem inalteradas. Em relação às escolhas dessas cargas, três critérios foram escolhidos resultando em três variantes da vizinhança N_5 , as quais denotaremos por N_5^a , N_5^b e N_5^c . A execução da busca local de acordo com cada uma dessas vizinhanças é repetida conforme o número de possibilidades para as escolhas que são explicadas a seguir.

Figura 19 – Exemplo de soluções vizinhas de acordo com a vizinhança N_5^a .

Fonte: Elaborada pelo autor.

Na vizinhança N_5^a , a escolha das cargas leva em consideração o instante de término de carregamento. Para isso, primeiramente, é escolhida como referência a doca com maior número de cargas de saída designadas. Para cada carga de saída nesta doca, toma-se uma carga de saída em cada uma das outras docas com tempo de término do carregamento mais próximo ao tempo de término do carregamento da carga na doca referência. Assim, o número de cargas da doca de referência define o número de repetições da busca local. Na Figura 19,

são ilustradas duas soluções vizinhas com base na vizinhança N_5^a e nas cargas escolhidas que estão circuladas em vermelho. Note que, tanto as rotas associadas às cargas escolhidas como a ordem de processamento das cargas são diferentes. Além disso, se as cargas escolhidas forem designadas para docas diferentes, as soluções também são soluções vizinhas de acordo com N_5^a .

Figura 20 – Possibilidades de escolhas para as buscas locais com a vizinhança N_5^b .

Fonte: Elaborada pelo autor.

No caso de N_5^b , as cargas são escolhidas considerando as posições em que elas são processadas. Assim, a busca local é executada considerando todas as cargas que são processadas na primeira posição, depois todas que são processadas na segunda posição e assim por diante. A busca local com a vizinhança N_5^b é executada somente se tiver mais de uma carga na mesma posição em docas diferentes. Na Figura 20, por exemplo, seriam 3 execuções da busca local considerando as 3 possibilidades de escolhas.

Figura 21 – Possibilidades de escolhas para as buscas locais com a vizinhança N_5^c .

Fonte: Elaborada pelo autor.

Na vizinhança N_5^c , as cargas são escolhidas intercalando duas posições consecutivas. Para melhor compreensão, exemplificamos as possibilidade de escolhas para um caso de solução ilustrada na Figura 21. Cada trio de cargas circuladas com linhas pontilhadas de mesma cor representa uma escolha. Por exemplo, as cargas A, E e C circuladas pela linha vermelha pontilhada e as cargas D, B e F circuladas pelas linha azul pontilhada representam duas escolhas com as posições 1 e 2 intercaladas. As outras escolhas são com as posições 2 e 3, depois com as posições 3 e 4 e assim sucessivamente. A direita na Figura 21, estão listadas todas as possibilidades de escolhas para o exemplo de solução ilustrada.

5.4 Experimentos computacionais

Os experimentos computacionais foram realizados utilizando o *solver* Gurobi (versão 8.1) na sua configuração padrão em um computador Intel Core i7-2600 2,40Gz com 15,6 GB (Linux 18.04 LTS). A FO-VND foi implementada em linguagem C++. Usamos os mesmos conjuntos de instâncias utilizados nos experimentos computacionais do Capítulo 3. Além disso, para avaliar as soluções obtidas pelo método proposto, elas são comparadas com as soluções obtidas pelo *solver* utilizando o Modelo M2 assim como no Capítulo 4, lembrando que, neste caso, o tempo limite foi de 1800 segundos para os conjuntos de instâncias com 15 e 25 clientes e de 2 horas para o conjunto de instâncias com 40 clientes.

Os experimentos foram realizados em duas fases. Na Fase 1, avaliamos duas versões do método. Na versão FO-VND1, são utilizadas buscas locais com as 7 vizinhanças na ordem em que são definidas nas seções anteriores. E, na versão FO-VND2, eliminamos a busca local com a vizinhança N_2 , porém incorporamos os movimentos da vizinhança N_2 nas vizinhanças N_3 , N_4 , N_5^a , N_5^b e N_5^c , ou seja, soluções com diferenças na designação e na ordenação das cargas de entrada também são consideradas soluções vizinhas. Na Fase 2, avaliamos algumas combinações das vizinhanças para as buscas locais com a melhor versão da FO-VND identificada na Fase 1.

5.4.1 Resultados computacionais - Fase 1

Os resultados computacionais da Fase 1 são resumidos nas Tabelas 17 – 22. Nas Tabelas 17 e 18, são reportados o número de instâncias em que a solução obtida pela FO-VND tem o valor da função objetivo menor ($VF^<$) e número de instâncias em que esse valor é igual ($VF^=$) ao valor obtido ao resolvê-las utilizando o Modelo M2 com o *solver* no tempo limite estabelecido. Também apresentamos os valores mínimo (D^{min}) e máximo (D^{max}), a média ($\bar{D}$) e o desvio padrão (D^σ) em relação às diferenças percentuais (D) entre as soluções da meta-heurística e do *solver* que são calculadas da seguinte forma:

$$D = \frac{SOL^{M2} - SOL^{FO-VND}}{SOL^{M2}} \times 100$$

onde SOL^{M2} é a solução obtida pelo *solver* ao resolver o Modelo M2 no tempo limite estabelecido e SOL^{FO-VND} é a solução obtida pela FO-VND. Ressaltamos que todas as soluções obtidas pela FO-VND são factíveis e as informações sobre os *gaps* das soluções obtidas pelo *solver* estão disponíveis no Apêndice B. Além disso, mais detalhes sobre as soluções obtidas pela FO-VND também estão disponíveis no Apêndice D.

No Cenário 1, conforme os resultados apresentados na Tabela 17, verificamos que as duas versões da metaheurística encontram soluções próximas ou melhores que as soluções obtidas pelo *solver* ao resolver o modelo M2 para os três conjuntos de instâncias, principalmente, a FO-VND2 cujo desempenho foi melhor no geral. Vale lembrar que cada conjunto é composto por 36 instâncias. A vantagem da FO-VND2 está possivelmente relacionada às buscas locais que são mais amplas ao considerar as decisões de designação e de ordenação nas docas de entrada

Tabela 17 – Desempenho da duas versões da FO-VND - Cenário 1.

		$\mathbf{VF}^{<}$	$\mathbf{VF}^{=}$	$\mathbf{D}^{min}$	$\mathbf{D}^{max}$	$\mathbf{\bar{D}}$	$\mathbf{D}^{\sigma}$
FO-VND1	15 clientes	1	8	-5,8	0,5	-0,9	1,3
	25 clientes	7	0	-8,2	5,5	-1,3	2,2
	40 clientes	26	0	-3,9	17,2	3,1	5,0
		11	3	-6,0	7,7	0,3	2,8
FO-VND2	15 clientes	3	17	-1,0	0,7	-0,2	0,4
	25 clientes	18	1	-3,5	4,7	-0,3	1,7
	40 clientes	18	0	-6,0	13,5	1,4	4,5
		13	6	-3,5	6,3	0,3	2,2

Fonte: Elaborada pelo autor.

e de saída simultaneamente, privilegiando as decisões no *cross-dock* que são mais relevantes no Cenário 1. No caso do conjunto de instâncias com 15 clientes, em que as soluções obtidas pelo *solver* apresentam *gap* médio de 1,2% sendo que 27 das 36 instâncias são soluções ótimas, as duas versões da FO-VND obtêm soluções com diferenças percentuais menores que 1% em média, sendo que a FO-VND1 e a FO-VND2 obtêm soluções com mesmo valor para 8 e 17 instâncias, respectivamente, das 27 com soluções ótimas. Para o conjunto de instâncias com 25 clientes, em que o *gap* das soluções obtidas pelo *solver* é em média 7,1%, a FO-VND1 obtém soluções até 5,5% melhores para 7 instâncias e a FO-VND2 obtém soluções até 4,7% melhores para 18 instâncias. Já para as instâncias com 40 clientes, a FO-VND1 teve melhor desempenho, apresentando solução com diferença de 3,9% no pior caso e de 17,2 no melhor caso, enquanto que FO-VND2 obteve, no pior caso, solução com 6,0% de diferença e, no melhor caso, uma diferença de 13,5%. Em resumo, do ponto de vista da qualidade da soluções, consideramos que a FO-VND2 tem uma melhor performance principalmente porque apresenta uma menor discrepância em relação à qualidade das solução de acordo com os valores mínimos e desvio padrão apresentados para os conjuntos de instâncias com 15 e 25 clientes e, no caso do conjunto com 40 clientes, a FO-VND1 teve melhor performance.

Tabela 18 – Desempenho das duas versões da FO-VND - Cenário 2.

		$\mathbf{VF}^{<}$	$\mathbf{VF}^{=}$	$\mathbf{D}^{min}$	$\mathbf{D}^{max}$	$\mathbf{\bar{D}}$	$\mathbf{D}^{\sigma}$
FO-VND1	15 clientes	3	10	-7,7	3,0	-1,3	2,6
	25 clientes	24	1	-15,7	8,0	1,0	4,0
	40 clientes	35	9	-2,0	27,1	13,1	6,0
		21	4	-8,5	12,7	4,2	4,2
FO-VND2	15 clientes	2	18	-6,0	2,4	-0,4	1,3
	25 clientes	22	1	-8,8	7,6	0,3	3,5
	40 clientes	33	0	-11,2	17,1	5,8	6,6
		18	6	-8,7	9,0	1,9	3,8

Fonte: Elaborada pelo autor.

No caso do Cenário 2, as soluções obtidas pelo *solver* (Modelo M2) estão mais distantes da otimalidade com *gaps* iguais, em média, a 6,7%, 20,4% e 38,6%, respectivamente, para os conjuntos de instâncias com 15, 25 e 40 clientes. Em geral, ao observar os resultados na Tabela 18, a FO-VND2 tem uma performance melhor para o conjunto com 15 clientes e a FO-VND1 tem uma performance menor para os conjuntos de instâncias com 25 e 40 clientes. No caso das instâncias com 15 clientes, a FO-VND1 e a FO-VND2, respectivamente, forneceram as mesmas soluções obtidas com o *solver* para 10 e 18 instâncias. Além disso, as duas versões fornecem soluções com diferenças de até 7,7% no pior caso e de até 3% no melhor caso. Para as instâncias com 25 clientes, a FO-VND1 obtém soluções melhores (entre 0,4% e 8,0%) para 24 instâncias e FO-VND2 obtém soluções melhores (entre 0,2% e 7,6%) para 22 instâncias. Já para as instâncias com 40 clientes, tivemos um desempenho significativamente melhor que o *solver* com destaque para a FO-VND1 que forneceu soluções entre 4,5% e 27,8% melhores para 26 das 27 instâncias para as quais o *solver* consegue obter pelo menos uma solução, apenas uma solução foi 2,0% pior. No caso das outras 9 instâncias, para as quais o *solver* não obteve solução, as soluções iniciais obtidas com a heurística construtiva tiveram uma melhora média de 26,5% e 19,7%, respectivamente, com a FO-VND1 e a FO-VND2.

Tabela 19 – Comparação entre a solução inicial (heurística construtiva) com a melhor solução obtida pela meta-heurística.

		Cenário 1			Cenário 2		
		M^{min}	M^{max}	$\bar{M}$	M^{min}	M^{max}	$\bar{M}$
FO-VND1	15 clientes	3,4	38,8	15,8	6,0	32,2	17,2
	25 clientes	7,2	59,2	23,3	7,1	36,7	19,9
	40 clientes	8,3	68,1	31,2	13,6	42,0	26,7
		6,3	55,4	23,4	8,9	37,0	21,3
FO-VND2	15 clientes	3,4	39,3	16,4	8,2	32,2	18,0
	25 clientes	8,2	59,2	24,0	7,1	35,4	19,3
	40 clientes	8,8	62,9	29,6	8,7	39,4	20,4
		6,8	53,8	23,3	8,0	35,7	19,2

Para avaliar as melhorias obtidas com a meta-heurística em relação a solução inicial com a qual o método é iniciado, reportamos, na Tabela 19, a menor (M^{min}), a maior (M^{max}) e a média ($\bar{M}$) das diferenças percentuais M calculadas da seguinte forma:

$$M = \frac{SOL^{HC} - SOL^{FO-VND}}{SOL^{HC}} \times 100$$

onde SOL^{HC} é a solução obtida pela heurística construtiva e SOL^{FO-VND} é a solução obtida pela FO-VND.

De acordo com a Tabela 19, tivemos melhorias significativas com diferenças de até 68% no Cenário 1 e de até 42% no Cenário 2. Importante ressaltar que, mesmo no caso em que as soluções dadas pela heurística construtiva são inviáveis por apresentarem tempo extra, as soluções obtidas com a FO-VND são todas factíveis, apresentando tempo extra nulo. Observamos

também que com a versão FO-VND1 obtivemos diferenças percentuais em média um pouco maiores que a versão FO-VND2.

Tabela 20 – Médias dos tempos de solução (em segundos).

	Cenário 1			Cenário 2		
	FO-VND1	FO-VND2	<i>Solver</i>	FO-VND1	FO-VND2	<i>Solver</i>
15 clientes	22	30	616	48	108	1541
25 clientes	243	683	1794	285	702	1800
40 clientes	1478	1729	7200	1545	1380	7200

Fonte: Elaborada pelo autor.

Na Tabela 20, são apresentados as médias dos tempos de solução das duas versões da FO-VND e do *solver*. Numa primeira comparação entre as duas versões da FO-VND, verificamos que o tempo médio gasto da FO-VND2 é maior, com exceção no Cenário 2 para as instâncias com 40 clientes em que ele foi um pouco menor que o da FO-VND1. Essa diferença está relacionado à ampliação da estrutura das vizinhanças com a liberação das decisões de designação e de ordenação das cargas de entrada que reflete no tempo de solução dos subproblemas resolvidos durante as buscas locais. Ao compararmos os tempos das duas versões da FO-VND com o *solver*, verificamos que os tempos do *solver* são bem maiores com diferenças ainda mais notáveis em relação à FO-VND1. Diante da boa qualidade das soluções obtidas e dos ganhos em termos de tempo de solução, concluímos que as duas versões da FO-VND, principalmente a FO-VND1, apresentam uma boa performance quando comparado ao *solver*.

A seguir, nas Tabelas 21 e 22, respectivamente, para o Cenário 1 e 2, reportamos a média da frequência em que é obtida uma melhoria ao efetuar uma busca local para cada uma das vizinhanças. Lembrando que, no caso da FO-VND2, a busca local com a vizinhança N_2 foi eliminada e as vizinhanças N_3 , N_4 , N_5^a , N_5^b e N_5^c foram ampliadas de forma que as soluções vizinhas também tenham diferenças nas decisões de designação e de ordenação das cargas de entrada.

Da maneira como foram definidas as vizinhanças, as buscas locais com N_1 , N_2 e N_3 são distintas em relação às decisões consideradas, enquanto a vizinhança N_4 é uma extensão da N_3 em que duas docas são consideradas. Na Tabela 21, melhorias com essas quatro buscas são mais frequentes de uma forma geral, sendo relevantes no processo de refinamento da solução. Observe que, na FO-VND2, em que a busca local com a N_2 é eliminada e as vizinhanças N_3 e N_4 são ampliadas, conforme explicado anteriormente, percebemos uma distribuição no total de frequência. Por outro lado, embora o desempenho da FO-VND2 tenha apresentado uma performance com desvio padrão menor e uma média maior, FO-VND2 apresenta um tempo de solução aproximadamente em média 63% maior que o tempo de solução apresentado pela FO-VND1. Conforme já mencionado, esse aumento é em função do aumento de tempo da resolução dos subproblemas resolvidos para as buscas locais. Note também que as buscas locais

Tabela 21 – Frequência média de melhorias de acordo com as vizinhanças - Cenário 1.

		N_1	N_2	N_3	N_4	N_5^a	N_5^b	N_5^c
FO-VND1	15 clientes	2,7	0,3	1,3	1,8	0,2	0,1	0,1
	25 clientes	7,2	1,9	1,4	5,6	0,3	0,1	0,3
	40 clientes	11,1	6,0	4,3	23,1	2,3	0,3	0,1
		7,0	2,7	2,4	10,1	0,9	0,2	0,2
FO-VND2	15 clientes	2,7	-	1,5	2,6	0,3	0,1	<0,1
	25 clientes	6,6	-	4,0	4,9	0,9	0,2	0,2
	40 clientes	7,8	-	9,7	8,7	<0,1	<0,1	0,1
		7,1	-	6,1	6,8	0,2	0,0	0,1

Fonte: Elaborada pelo autor.

com N_5^a , N_5^b e N_5^c tiveram as médias de frequência menores que 1.

Tabela 22 – Frequência média de melhorias de acordo com as vizinhanças - Cenário 2.

		N_1	N_2	N_3	N_4	N_5^a	N_5^b	N_5^c
FO-VND1	15 clientes	3,2	0,9	2,9	2,9	0,1	<0,1	<0,1
	25 clientes	6,3	2,9	5,1	5,1	0,2	<0,1	0,1
	40 clientes	12,0	4,7	10,4	12,3	0,2	0,1	0,1
		7,1	2,8	6,1	6,8	0,2	<0,1	0,1
FO-VND2	15 clientes	2,4	-	3,4	2,8	0,3	<0,1	0,1
	25 clientes	4,8	-	7,3	4,0	0,1	0,1	0,3
	40 clientes	6,8	-	13,0	0,8	<0,1	<0,1	<0,1
		4,6	-	7,9	2,5	0,1	<0,1	0,1

Fonte: Elaborada pelo autor.

No caso de Cenário 2, conforme as médias das frequências na Tabela 22, temos basicamente o mesmo comportamento do Cenário 1. Em particular, para o conjunto de instâncias com 40 clientes, temos uma redução na frequência de melhorias com as vizinhanças N_1 e N_4 quando usamos a versão FO-VND2.

De forma geral, no desenvolvimento de métodos heurísticos, procuramos métodos que forneçam boas soluções com tempos de solução razoáveis. Conforme os resultados apresentados nas duas últimas tabelas, verificamos que algumas buscas locais para algumas vizinhanças apresentaram uma baixa frequência de melhorias. Diante disso, é interessante investigar o impacto da eliminação dessas buscas sobre a qualidade e o tempo de solução ou o efeito de algumas combinações entre as buscas locais propostas. É com foco nessa investigação que executamos uma segunda fase de testes cujos resultados estão apresentados a seguir.

5.4.2 Resultados computacionais - Fase 2

Na Fase 2, avaliamos algumas versões da FO-VND variando a combinação de suas vizinhanças. Conforme o desempenho apresentado, escolhemos a FO-VND2 para o Cenário 1 e a FO-VND1 para o Cenário 2. A escolha das vizinhanças foi baseada nas frequências médias de melhorias obtidas na Fase 1 dos testes e reportadas nas Tabelas 21 e 22. Primeiro, avaliamos a FO-VND com as vizinhanças que apresentaram as maiores frequências médias de melhorias que foram as vizinhanças N_1 , N_3 e N_4 no Cenário 1 e as vizinhanças N_1 , N_2 , N_3 e N_4 no Cenário 2. Depois, incluímos também a vizinhança N_5^a a partir da qual as vizinhanças apresentaram baixas frequências médias de melhorias. Note que, dependendo das escolhas, as vizinhanças N_5^a , N_5^b e N_5^c podem representar uma mesma busca local e todas tiveram baixas frequências médias de melhorias. Para analisá-las, retiramos a busca local com a N_4 e incluímos as buscas locais com as vizinhanças N_5^a , N_5^b e N_5^c separadamente, resultando em mais três versões da FO-VND.

Tabela 23 – Resultados para algumas combinações das vizinhanças - Cenário 1.

		VF<	VF=	D^{min}	D^{max}	$\bar{D}$	D^σ	T(s)
Todas	15 clientes	3	17	-1,0	0,7	-0,2	0,4	30
	25 clientes	18	1	-3,5	4,7	-0,3	1,7	683
	40 clientes	18	0	-6,0	13,5	1,4	4,5	1729
		13	6	-3,5	6,3	0,3	2,2	814
N_1, N_3, N_4	15 clientes	3	13	-3,2	0,6	-0,4	0,7	24
	25 clientes	12	1	-6,3	4,7	-1,3	2,4	281
	40 clientes	17	0	-11,8	13,5	0,5	5,3	1476
		11	5	-7,1	6,2	-0,4	2,8	593
N_1, N_3, N_4, N_5^a	15 clientes	3	15	-1,0	0,7	-0,2	0,4	28
	25 clientes	17	1	-4,7	4,7	-0,6	2,0	426
	40 clientes	17	0	-11,8	13,5	0,8	5,1	1576
		12	5	-5,9	6,3	0,0	2,5	677
N_1, N_3, N_5^a	15 clientes	1	4	-5,5	0,5	-1,6	1,6	16
	25 clientes	5	0	-6,7	3,0	-2,1	2,2	339
	40 clientes	7	0	-12,8	12,9	-3,2	5,0	721
		4	1	-8,3	5,5	-2,3	3,0	359
N_1, N_3, N_5^b	15 clientes	1	2	-4,3	0,5	-1,6	1,3	16
	25 clientes	4	0	-6,7	1,3	-2,5	2,1	285
	40 clientes	8	0	-13,6	12,9	-3,3	5,1	751
		4	1	-8,2	4,9	-2,4	2,8	351
N_1, N_3, N_5^c	15 clientes	2	2	-4,0	1,0	-1,0	1,2	20
	25 clientes	8	0	-8,7	1,8	-1,2	2,2	472
	40 clientes	11	0	-11,8	12,9	-2,0	5,0	947
		7	1	-8,2	5,3	-1,4	2,8	479

Fonte: Elaborada pelo autor.

Assim como nas Tabelas 17 e 18, nas Tabelas 23 e 24, respectivamente, para os Cenários 1 e 2, reportamos o número de instâncias em que a solução obtida pela FO-VND tem o valor da

função objetivo menor ($VF^<$) e número de instâncias em que esse valor é igual ($VF^=$) ao valor obtido pelo *solver* no tempo limite estabelecido. Também apresentamos os valores mínimo (D^{min}) e máximo (D^{max}), a média ($\bar{D}$) e o desvio padrão (D^σ) em relação às diferenças percentuais entre as soluções da metaheurística e do *solver* calculadas de forma análoga à Fase 1. Também incluímos o tempo médio de solução em segundos (T) na última coluna a direita. Na primeira coluna a esquerda, especificamos as vizinhanças na ordem em que são utilizadas para as buscas locais em cada versão da FO-VND. Os valores em negritos representa as médias dos valores apresentados nas tabelas para cada uma das informações.

Tabela 24 – Resultados para algumas combinações das vizinhanças - Cenário 2.

		$VF^<$	$VF^=$	D^{min}	D^{max}	$\bar{D}$	D^σ	T(s)
Todas	15 clientes	3	10	-7,7	3,0	-1,3	2,6	48
	25 clientes	24	1	-15,7	8,0	1,0	4,0	285
	40 clientes	35	0	-2,0	27,1	13,1	6,0	1545
		21	4	-8,5	12,7	4,2	4,2	626
N_1, N_2, N_3, N_4	15 clientes	3	9	-7,8	3,0	-1,6	2,7	39
	25 clientes	23	1	-15,7	8,0	0,8	4,0	190
	40 clientes	35	0	-2,0	27,1	12,6	6,0	1289
		20	3	-8,5	12,7	4,0	4,2	506
$N_1, N_2, N_3, N_4, N_5^a$	15 clientes	3	10	-7,7	3,0	-1,3	2,6	45
	25 clientes	24	1	-15,7	8,0	1,0	4,0	232
	40 clientes	35	0	-2,0	27,1	12,8	6,0	1383
		21	4	-8,5	12,7	4,1	4,2	553
N_1, N_2, N_3, N_5^a	15 clientes	2	1	-12,9	3,0	-4,1	3,4	7
	25 clientes	6	1	-7,7	8,3	-0,4	3,5	109
	40 clientes	29	0	-4,5	22,2	5,7	6,8	582
		12	1	-8,4	11,2	0,4	4,6	233
N_1, N_2, N_3, N_5^b	15 clientes	2	1	-17,1	3,0	-4,6	4,1	5
	25 clientes	14	1	-7,7	6,0	-0,9	3,2	83
	40 clientes	31	0	-20,3	28,1	6,5	8,7	570
		16	1	-15,0	12,4	0,3	5,3	219
N_1, N_2, N_3, N_5^c	15 clientes	1	3	-11,4	2,4	-3,9	3,1	7
	25 clientes	14	1	-16,1	8,3	-1,4	4,5	87
	40 clientes	32	0	-3,5	27,9	7,4	6,7	669
		16	1	-10,4	12,9	0,7	4,8	254

Fonte: Elaborada pelo autor.

Analisando a Tabela 23, percebemos que as vizinhanças N_1 , N_3 e N_4 são bastantes relevantes no processo de melhoria da solução. E, quando elas são usadas junto com a vizinhança N_5^a , percebemos uma pequena melhora ao observarmos as médias das diferenças percentuais e é a versão que mais se aproxima da performance quando todas as vizinhanças são usadas, além de ter o tempo médio de solução reduzido em aproximadamente 13%. Note também que, quando usamos as vizinhanças N_5^a , N_5^b e N_5^c excluindo a N_4 , tivemos em geral resultados piores.

De acordo com os resultados para o Cenário 2 na Tabela 24, observamos que as versões com as vizinhanças N_1 , N_2 , N_3 e N_4 (com e sem N_5^a) com performance muito similar à performance da versão em que todas as vizinhanças são utilizadas. Neste caso, ao excluirmos as buscas locais com as vizinhanças N_5^b e N_5^c , além de não termos perdas significativas na qualidade das soluções, reduzimos um pouco o tempo médio de solução. Em geral, as outras versões em que são testadas as vizinhanças N_5^a , N_5^b e N_5^c separadamente e sem N_4 apresentam também desempenho piores.

5.5 Conclusões

Conforme apontaram os resultados a meta-heurística proposta teve um bom desempenho, em especial, para as instâncias maiores (40 clientes) em que, além de ganhos em tempo de solução, foi possível obter soluções melhores que as obtidas pelo *solver* ao utilizando diretamente o Modelo M2 para quase todas as instâncias e com diferenças significativas de até 17,2% no Cenário 1 e 27,1% no Cenário 2. Além disso, essas soluções melhores são obtidas em um tempo que é, aproximadamente, um quarto do tempo utilizado pelo *solver*. Também é importante ressaltar que todas as soluções iniciais obtidas pela heurística construtiva e que eram infactíveis foram factibilizadas.

Ao analisarmos as frequências de melhorias obtidas com as buscas locais utilizando as diferentes vizinhanças, verificamos que algumas não eram significativas no processo de melhoria da solução. De acordo com os resultados na Fase 2, identificamos que as vizinhanças N_1 , N_2 , N_3 e N_4 , no caso da FO-VND1, e as vizinhanças N_1 , N_3 e N_4 , no caso da FO-VND2, são as que mais contribuem para o refinamento das soluções. Apenas com essas vizinhanças, é possível obter boas soluções, principalmente para as instâncias com 40 clientes. Logo, como esperado, a meta-heurística desenvolvida é indicada para instâncias de maior dimensão.

No próximo capítulo, abordaremos uma matheurística para o problema que também pode ser incorporada como mais um passo no refinamento da solução obtida pela meta-heurística.

MATHEURÍSTICA: *PROXIMITY SEARCH*

A análise dos resultados obtidos ao resolver instâncias do problema estudado utilizando um *solver* comercial de otimização apresentada no Capítulo 3 confirmou que as decisões de descarregamento, de consolidação, de carregamento e de roteamento são fortemente relacionadas e interdependentes. Como já era esperado, também mostrou que o problema fica mais difícil de resolver à medida que sua dimensão aumenta. Diante disso, acreditamos que um método de solução que considere todas as decisões de forma conjunta poderia ser uma boa opção para tratar instâncias maiores do problema. Assim, escolhemos a matheurística *Proximity Search* para buscar melhores soluções para o problema.

A matheurística *Proximity Search* (PS) foi introduzida por Fischetti e Monaci (2014) tendo como foco principal resolver problemas de programação inteira com variáveis binárias (PIB). Os autores investigaram o efeito de trocar a função objetivo de um PIB por uma função de proximidade com a melhor solução obtida até um determinado momento. Mais especificamente, a distância é medida de acordo com a distância de Hamming. Além disso, é adicionada uma restrição de corte que impõe que a função objetivo associada à próxima solução seja θ melhor que a solução incumbente e, assim, gera-se uma sequência de soluções melhoradas. De acordo com os autores, conforme verificado em experimentos computacionais apresentados, a modificação na função objetivo tende a potencializar o desempenho heurístico do *solver* de otimização utilizado para resolver o problema. Berthold, Lodi e Salvagnin (2019) observam que a PS é, na verdade, uma combinação de elementos de busca local com elementos da matheurística *Feasibility Pump*.

Em uma breve revisão bibliográfica, identificamos alguns trabalhos que utilizam a matheurística como parte do seu método de solução. Por exemplo, Fischetti e Monaci (2016) e Fischetti, Fischetti e Monaci (2016) estudam o problema de posicionamento das turbinas eólicas com o objetivo de maximizar a geração de energia. Os autores propõem um método de solução que combinam *ad-hoc* heurísticas com a PS. Nos dois trabalhos, os métodos de solução com a PS tiveram uma melhor performance quando comparado ao *solver* CPLEX e às heurísticas sem o

uso da PS. Já [Álvarez-Miranda et al. \(2014\)](#) resolvem um problema de redes (*single-commodity robust network design problem*) utilizando uma matheurística composta por três fases sendo uma delas a PS. O desempenho do método é similar ao CPLEX para instâncias pequenas, porém, para as instâncias grandes a matheurística apresenta melhores soluções.

A seguir, apresentamos as adaptações feitas na matheurística *Proximity Search* para tratarmos o problema estudado. Depois, os experimentos computacionais realizados são detalhados.

6.1 Algoritmo da matheurística

O Algoritmo 3 resume os passos da matheurística implementada. A partir de uma dada solução inicial $\hat{X}$, a cada iteração, o método consiste em obter sua solução X tal que $f(\hat{X}) - f(X)$ seja pelo menos θ ($\theta > 0$), onde f é a função objetivo (3.1) definida no Capítulo 3. A sequência de soluções são obtidas com a resolução de um modelo de programação linear inteira mista auxiliar definido a partir do Modelo M2. Basicamente, o **Modelo Auxiliar (MA)** é o Modelo M2 com uma restrição de corte e uma nova função objetivo definida a partir da solução incumbente. Assim, se uma solução melhor é encontrada, o Modelo Auxiliar é atualizado, e, caso contrário, apenas o valor de θ é reduzido.

Algoritmo 3 – *Proximity Search*

Entrada: Solução inicial ($\hat{X}$), dados do problema e tempo máximo do método (TH^{max}).

Saída: Melhor solução encontrada.

- 1: **enquanto** $tempo < TH^{max}$ **faça**
 - 2: Construa o **Modelo Auxiliar (MA)** a partir de $\hat{X}$ e de θ .
 - 3: Respeitando o **Critério de parada**, resolva **MA**.
 - 4: Faça X^* receber a melhor solução obtida ao resolver **MA**.
 - 5: **se** X^* for melhor que $\hat{X}$ **então**
 - 6: Refine X^* resolvendo o Modelo M2 com as variáveis binárias de X^* fixas.
 - 7: Faça $\hat{X} = X^*$ refinada.
 - 8: **senão**
 - 9: Reduza o valor de θ .
-

A seguir, apresentamos os processos de obtenção da solução inicial e de construção do Modelo Auxiliar e os critérios de parada adotados na resolução do Modelo Auxiliar.

6.2 Solução inicial

Testes computacionais usando instâncias com 40 clientes, por exemplo, mostraram que pode levar bastante tempo para obter uma solução viável ou, às vezes, nem é possível obtê-la em duas horas ao resolver o Modelo M2 utilizando um *solver* comercial de otimização Gurobi conforme já mencionado nos Capítulos 4 e 5. Logo, buscamos uma estratégia alternativa para obter uma solução inicial a partir do próprio modelo de forma mais rápida, especialmente para as

instâncias com grande dimensão. Para isso, restringimos o espaço de soluções fixando algumas variáveis x_{ij} em 0 de acordo com os critérios definidos a seguir e, depois, resolvemos o Modelo M2.

As variáveis x_{ij} são fixadas em 0 se, entre os clientes i e j , pelo menos um dos casos a seguir ocorre.

1. $A_i + Q_iST + TT_{ij} > B_j$, ou seja, quando não é possível atender o cliente i e, na sequência, atender o cliente j respeitando a janela de tempo de ambos.
2. $A_j - \left(\frac{A_i+B_i}{2} + Q_iST + TT_{ij}\right) > 0$, ou seja, dado que o cliente i é atendido no instante médio de sua janela de tempo, não é possível atender j dentro da sua janela de tempo sem que haja um tempo de espera.
3. $\left(\frac{A_i+B_i}{2} + Q_iST + TT_{ij}\right) - B_j > 0$, tem a mesma ideia do caso 2, porém sem que haja atraso.

Note que Q_iST é o tempo gasto para descarregar Q_i paletes do cliente i .

No caso de instâncias com 40 clientes, essa estratégia de fixação de variáveis não foi suficiente para encontrar rapidamente uma solução viável para 39 das 72 instâncias (11 do Cenário 1 e 28 do Cenário 2). A maioria dessas instâncias têm janelas de tempo mais largas, logo poucas variáveis puderam ser fixadas em 0. Assim, definimos mais um critério para fixar as variáveis x_{ij} que é aplicado após os critérios anteriores.

O novo critério é definido da seguinte maneira. Primeiramente, para cada cliente i , é calculada a média dos tempos de viagem do cliente i ($\overline{TV}_i$) para todos os clientes cuja visita poderá suceder a visita do cliente i de acordo com os critérios anteriores definidos, ou seja,

$$\overline{TV}_i = \sum_{j \in \mathcal{N}_i^*} \frac{TT_{ij}}{|\mathcal{N}_i^*|}$$

onde $\mathcal{N}_i^*$ é o conjunto dos clientes que podem ser visitados depois da visita no cliente i após serem aplicados os critérios 1, 2 e 3. Assim, para cada cliente i , fixamos as variáveis x_{ij} em 0 se $TT_{ij} < \overline{TV}_i/2$ para todo j em $\mathcal{N}_i^*$.

Finalmente, depois de fixar as variáveis x_{ij} em 0 de acordo com os critérios apresentados, resolvemos o Modelo M2 com tempo limite de 120 segundos ou até atingir um *gap* menor ou igual a 50%. A solução obtida é usada como solução inicial para a PS adaptada. Mesmo utilizando estes quatro critérios, não foi possível obter uma solução em 120 segundos para quatro instâncias do Cenário 1 e para cinco do Cenário 2.

6.3 Modelo Auxiliar

O **Modelo Auxiliar (MA)** é obtido a partir do modelo M2 com a adição de uma restrição de corte e a substituição da função objetivo. Ambas obtidas a partir de uma dada solução inicial viável $\widehat{X} = (\widehat{x}, \widehat{z}^{in}, \widehat{z}^{out}, \widehat{w}^{in}, \widehat{w}^{out}, \widehat{et}, \widehat{lt}, \widehat{dt}^{max})$. A variável $\widehat{y}$ foi omitida aqui, porque ela é consequência dos valores assumidos pelas outras variáveis e, por isso, não a incluímos na função objetivo.

O conjunto de restrições de MA inclui todas as restrições do Modelo M2 e a restrição de corte (6.1).

$$f(x, et, lt, dt^{max}) \leq f(\widehat{x}, \widehat{et}, \widehat{lt}, \widehat{dt}^{max}) - \theta(1 - \zeta), \quad (6.1)$$

onde θ é uma dada tolerância e $\zeta \in [0, 1]$ é uma variável que relaxa a tolerância θ . Assim, se ζ assume o valor 1, a solução incumbente é uma solução viável para o Modelo MA e é dada como uma solução de partida para sua resolução pelo *solver*. Como a função objetivo f do Modelo M2 depende apenas de algumas componentes de X e de $\widehat{X}$, na restrição (6.1), escrevemos f somente em função das componentes utilizadas no cálculo do seu valor.

A função objetivo do modelo auxiliar visa minimizar a distância de Hamming (Δ) entre a solução inicial e a próxima solução a ser encontrada. A minimização dessa distância é escrita como:

$$\text{Min } \Delta = \Delta(x, \widehat{x}) + \Delta(z^{in}, \widehat{z}^{in}) + \Delta(z^{out}, \widehat{z}^{out}) + \Delta(w^{in}, \widehat{w}^{in}) + \Delta(w^{out}, \widehat{w}^{out}) + \text{BigM}\zeta, \quad (6.2)$$

onde $X = (x, z^{in}, z^{out}, w^{in}, w^{out}, et, lt, dt^{max})$ é a próxima solução a ser obtida e BigM é um número suficientemente grande de forma que a variável ζ seja forçada a assumir o menor valor possível. Assim, por exemplo, se ζ assume o valor 0,2 temos com certeza uma nova solução X tal que a diferença $f(\widehat{X}) - f(X)$ é pelo menos $0,8 \times \theta$ que $\widehat{X}$.

Para exemplificar, a distância de Hamming entre x e $\widehat{x}$ é calculada da seguinte forma:

$$\Delta(x, \widehat{x}) = \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{j \in \mathcal{N} \cup \{N+1\} | i \neq j, \widehat{x}_{ij}=0} x_{ij} + \sum_{i \in \mathcal{N} \cup \{0\}} \sum_{j \in \mathcal{N} \cup \{N+1\} | i \neq j, \widehat{x}_{ij}=1} (1 - x_{ij}).$$

Para as outras componentes, o cálculo é análogo. Aqui, consideramos $\text{BigM} = 3 \times (N + 2) \times (N + 2) + L \times F + N \times H + L \times L$ que é com certeza um número significativamente maior que Δ .

6.4 Critérios de parada para resolução de MA

Dada uma solução viável inicial e construído o Modelo MA, ele é resolvido utilizando o *solver* de otimização Gurobi. Como este é um problema de difícil solução, três critérios de parada foram considerados: (i) tempo limite; (ii) *gap* de otimalidade; e (ii) número de soluções factíveis encontradas. Esses critérios têm a função de acelerar a convergência da matheurística. O tempo é limitado porque, em geral, uma solução melhor é encontrada, mas a prova de sua

otimalidade demanda grande esforço computacional. No caso do *gap*, quando ele atinge 99%, significa que já obtemos uma solução melhor com uma diferença de pelo menos θ em relação a solução atual. E, no caso do número de soluções, o critério é ativado somente quando a tolerância θ atingir o menor valor. Nesse momento, quando o número de soluções limite é atingindo, mas ainda há tempo computacional disponível, a última solução obtida é utilizada para reiniciar a PS.

O refinamento da melhor solução inteira encontrada ao resolver o MA é feito a partir do Modelo M2 fixando as variáveis binárias de acordo com a solução encontrada. O refinamento permite ajustar os valores das variáveis contínuas da solução obtida visando melhorar esta solução. Nesse momento, verificamos se o ganho da solução é menor ou igual a 5% e, caso ele seja, reduzimos o θ . Este valor foi obtido em testes preliminares. Isso é feito, pois observamos que dificilmente na próxima iteração não seria obtida uma solução melhor se continuássemos com o mesmo θ e, assim, antecipamos a redução do θ para a próxima iteração.

Diferentemente de alguns trabalhos da literatura que atribuem valores baixos para θ , admitimos no início valores mais altos visando obter maiores ganhos no início. Para o nosso problema, esse caminho se demonstrou mais eficiente nos testes preliminares, provavelmente porque nossa solução de partida tem um *gap* de otimalidade alto. Também baseado nos testes preliminares, definimos os seguintes valores para os parâmetros. Consideramos $\theta = \lambda f(\hat{x}, \hat{et}, \hat{lt}, \hat{dt}^{max})$, com λ assumindo os seguintes valores 0,25, 0,20, 0,15, 0,10, 0,05, 0,01 e 0,001. Para os valores de λ de 0,25 a 0,10 o tempo limite para resolução do MA é inicialmente de 120 segundos e diminui 20 segundos a cada vez que λ reduz. Para os valores menores que 0,10 o tempo limite é fixado em 60 segundos, exceto se λ já estiver atingido seu valor mínimo, quando definimos o tempo limite igual ao tempo restante para atingir TH^{max} e impomos o critério que limita o número de soluções viáveis igual a 3. Assim, se a terceira solução viável não for melhor, aumentamos em 1 o limite do número de soluções factíveis e reiniciamos a PS com a última solução obtida e o tempo restante para atingir o TH^{max} . E, se a quarta solução viável também não for melhor, repetimos a PS aumentando novamente o limite para obtermos a quinta solução viável, e assim por diante. Quando uma solução encontrada for melhor, retornamos o limite de soluções viáveis para 3.

6.5 Experimentos computacionais

Os experimentos computacionais foram realizados usando o *solver* Gurobi versão 8.1 em um computador Intel Core i7-2600 2,40Gz com 15,6 GB (Linux 18.04 LTS). Os mesmos conjuntos de instâncias com 15, 25 e 40 clientes utilizados nos experimentos computacionais do Capítulo 3 são usados aqui. Configuramos o *solver* Gurobi para priorizar a obtenção de soluções viáveis impondo o parâmetro “MIPFocus = 1”. Considerando diferentes maneiras de obter a solução inicial, avaliamos duas versões da matheurística especificadas a seguir:

- PS1: a solução inicial dada é obtida de acordo com o procedimento definido na Seção 6.2

que utiliza o Modelo M2 fixando os valores de algumas variáveis.

- PS2: a solução inicial dada é a obtida pela heurística construtiva do Capítulo 4. Nesse caso, admitimos a solução heurística mesmo que ela não seja viável por apresentar um tempo extra para a conclusão das entregas e o retorno dos veículos ao *cross-dock* com a inclusão da variável t^{ex} . Essa variável representa o tempo extra penalizado no cálculo de $f(x, et, lt, dt^{max})$, análogo ao que é feito no Capítulo 5 para a meta-heurística. Assim, na primeira solução obtida em que a variável t^{ex} é nula, fixamos seu valor em 0.

Resolvemos as instâncias com o tempo limite de processamento (TH^{max}) de 1800 segundos, incluindo o tempo para obter a solução inicial. Em especial, para o conjunto de instâncias com 15 clientes, esse tempo foi limitado a 600 segundos. Para a avaliação das soluções obtidas, comparamos com as soluções fornecidas pelo *solver* ao resolver o Modelo M2 da mesma maneira como é realizado nos Capítulos 4 e 5. Lembrando que, no caso da resolução do Modelo M2 diretamente pelo *solver*, o tempo de solução foi limitado a 1800 segundos para os conjuntos de instâncias com 15 e 25 clientes e a 7200 segundos para o conjunto de instâncias com 40 clientes. Para cada instância, calculamos as diferenças percentuais (D) entre as soluções obtidas da seguinte forma:

$$D = \frac{SOL^{M2} - SOL^{PS}}{SOL^{M2}} \times 100$$

onde SOL^{M2} é a solução obtida pelo *solver* ao resolver o Modelo M2 no tempo limite estabelecido e SOL^{PS} é a solução obtida pela matheurística.

Nas Tabelas 25 e 26, estão resumidos os resultados dos experimentos, respectivamente, para os Cenários 1 e 2. São reportados os valores da função objetivo (VF) e os *gaps* em percentagem das soluções fornecidas pelo *solver* ao resolver o Modelo M2 e as diferenças percentuais (D) considerando as soluções de cada versão da matheurística. Além disso, nos espaços onde há um traço ao invés de um número está representada a situação em que o *solver* não obteve solução resolvendo o Modelo M2 ou a PS1 não obteve solução inicial de acordo com o procedimento explicado na Seção 6.2.

De acordo com a Tabela 25, notamos que as soluções obtidas pelas duas versões da matheurística estão próximas (menos de 0,9%) ou coincidem com a maioria das soluções ótimas das instâncias com 15 clientes. Lembrando que, neste caso, o tempo de processamento é praticamente igual ao tempo médio de solução do *solver* que é 616 segundos. Para as instâncias com 25 clientes, principalmente para as dos Grupos B e C, temos que os melhores resultados foram obtidos pela PS2 com ganhos de até 5,5%. Já para as instâncias com 40 clientes, as melhores soluções obtidas com as duas versões da matheurística apresentaram ganhos mais significativos de até 17%. Vale a pena ressaltar que, mesmo com as soluções infactíveis por conta do tempo extra apresentado, todas as soluções finais obtidas pelas duas versões da PS são factíveis. Em especial, com a PS1, não foi possível obter solução inicial para as instâncias R112-1B, R103-1C, R104-1C e R111-1C com o procedimento explicado na Seção 6.2. Importante

Tabela 25 – Resultados para as instâncias do Cenário 1.

Instância	15 clientes				25 clientes				40 clientes			
	Modelo M2		PS1	PS2	Modelo M2		PS1	PS2	Modelo M2		PS1	PS2
	VF	GAP	D	D	VF	GAP	D	D	VF	GAP	D	D
R101-1A	1090,0	0,0	0,0	0,0	1532,1	0,0	-0,6	-0,6	2256,5	17,3	1,2	5,6
R102-1A	1089,8	0,0	0,0	0,0	1500,0	0,7	-0,2	-0,1	2335,8	10,4	1,0	1,3
R103-1A	999,3	0,0	0,0	0,0	1488,2	2,4	0,0	-0,1	1989,1	6,1	-0,7	0,0
R104-1A	905,4	0,0	0,0	0,0	1397,3	2,8	0,3	0,3	2081,2	5,2	0,2	0,4
R105-1A	1119,3	0,0	0,0	0,0	1545,9	3,6	-9,5	-0,3	2289,0	11,9	-3,0	0,2
R106-1A	982,2	0,0	0,0	-0,3	1421,1	5,4	-1,3	-0,5	2293,1	10,7	2,2	-1,5
R107-1A	1062,9	0,0	0,0	0,0	1470,9	0,5	0,0	-0,2	1924,7	1,3	-0,1	-1,1
R108-1A	788,9	0,0	0,0	0,0	1475,2	2,3	0,0	-0,8	1877,8	3,4	-3,1	-3,2
R109-1A	855,5	0,0	0,0	0,0	1287,4	5,6	0,6	0,8	2129,0	5,2	-0,6	-3,2
R110-1A	1058,4	0,0	0,0	0,0	1439,2	0,9	-0,4	-0,6	1847,2	5,0	0,9	0,6
R111-1A	1042,6	0,0	0,0	-0,1	1406,4	6,4	0,1	-0,2	2034,1	2,7	-0,5	-0,2
R112-1A	943,2	0,0	0,0	0,0	1530,5	2,9	-1,0	-0,7	2079,6	2,4	0,3	0,4
R101-1B	916,2	5,5	0,2	0,2	1498,4	14,4	-2,3	1,2	2320,6	17,0	-5,9	6,0
R102-1B	1173,7	0,0	0,0	0,0	1532,6	4,4	0,3	0,0	2418,9	14,6	2,8	-5,9
R103-1B	994,6	0,0	0,0	0,0	1481,4	4,9	-0,2	-0,1	2185,5	12,1	6,2	4,9
R104-1B	880,4	0,0	0,0	0,0	1425,7	10,8	0,8	1,5	2121,8	6,0	1,4	-0,7
R105-1B	1048,4	0,0	0,0	0,0	1549,8	6,4	-0,6	0,6	2503,8	18,5	3,6	-5,3
R106-1B	982,6	0,0	0,0	0,0	1457,6	8,2	1,2	2,4	2379,8	13,4	3,4	4,0
R107-1B	1024,2	0,0	0,0	0,0	1480,2	4,9	-0,1	0,3	1963,5	10,2	3,2	3,2
R108-1B	806,8	0,0	0,0	0,0	1465,7	5,7	-0,4	0,2	1904,0	4,4	-0,8	-0,1
R109-1B	897,1	3,2	0,0	0,0	1331,8	13,0	2,8	2,4	2314,0	12,8	5,5	7,0
R110-1B	1042,2	0,0	-0,9	0,0	1451,2	6,9	-0,8	0,0	1862,7	7,4	-0,1	-1,8
R111-1B	1066,4	0,0	0,0	0,0	1753,2	6,3	0,2	1,5	2154,2	14,0	3,9	3,0
R112-1B	1032,8	0,0	0,0	0,0	1543,8	4,6	-0,4	-0,3	2100,1	7,6	-	1,4
R101-1C	1033,5	12,6	0,9	0,6	1688,6	26,3	1,7	5,5	2836,9	35,9	16,4	15,1
R102-1C	1154,4	0,8	0,0	-0,4	1572,5	12,8	0,5	0,5	2503,3	20,3	1,1	0,0
R103-1C	1005,9	0,0	0,0	0,0	1687,6	7,9	2,3	2,1	2268,1	13,4	-	6,8
R104-1C	878,9	0,0	0,0	0,0	1436,5	2,5	-0,7	0,0	2264,9	17,9	-	7,6
R105-1C	1367,0	13,4	0,2	0,5	1636,9	16,3	2,5	2,5	3064,6	34,7	17,0	14,1
R106-1C	1173,7	0,0	0,0	0,0	1475,6	6,2	0,4	-1,2	2484,7	18,9	2,8	-9,7
R107-1C	1019,8	0,0	0,0	0,0	1491,7	6,7	0,2	0,0	2169,6	22,8	11,5	11,4
R108-1C	823,5	1,3	0,0	0,0	1480,2	3,3	0,1	0,0	1939,6	5,8	1,2	1,6
R109-1C	915,1	0,0	0,0	0,0	1323,1	11,4	1,3	0,7	2949,9	29,2	11,1	0,8
R110-1C	985,3	2,9	0,3	0,3	1511,3	14,6	0,6	2,1	2007,6	12,9	3,6	7,7
R111-1C	1035,8	2,7	0,0	-0,1	1452,5	9,3	1,2	0,4	2187,6	13,8	-	5,3
R112-1C	998,0	1,8	0,0	0,0	1583,0	13,0	1,2	1,4	2090,4	10,8	0,7	1,4
Média		1,2	0,0	0,0		7,1	0,0	0,6		12,7	2,7	2,1

lembrar que, no caso das instâncias com 40 clientes, o tempo de processamento do *solver* na resolução do Modelo M2 foi maior (7200 segundos).

No caso do Cenário 2, para as instâncias com 15 clientes, como pode ser observado na Tabela 26, as soluções obtidas pelas PS1 e PS2 coincidem ou são próximas em relação às soluções obtidas resolvendo o Modelo M2, apenas em três casos tivemos soluções piores obtidas pela PS2 com diferenças maiores entre 2,9% e 17,6%. Vale a pena ressaltar que, neste caso, o tempo médio de processamento do *solver* é igual 1541 segundos maior que da *matheurística*. Para o conjunto de instâncias com 25 clientes, PS1 e PS2 obtêm soluções melhores para aproximadamente 75% das instâncias com ganhos entre 0,1% e 8,4%. Já para o conjunto com 40 clientes, PS1 e PS2 fornece soluções piores para apenas três e duas instâncias, respectivamente, com diferenças de 5,1% em média e, para as outras instâncias, tivemos soluções melhores com diferenças significativas entre 0,1% e 25,5%. Assim como acontece no Cenário 1, com o procedimento

Tabela 26 – Resultados para as instâncias do Cenário 2.

Instância	15 clientes				25 clientes				40 clientes			
	Modelo M2		PS1	PS2	Modelo M2		PS1	PS2	Modelo M2		PS1	PS2
	VF	GAP	D	D	VF	GAP	D	D	VF	GAP	D	D
R101-2A	682,4	0,0	0,0	0,0	958,1	17,4	-2,9	1,1	1587,6	30,9	-3,6	-1,5
R102-2A	610,5	18,7	-2,3	-0,4	843,8	15,1	-1,1	-1,9	-	-	-	-
R103-2A	633,4	5,2	0,0	0,0	801,8	22,8	1,7	2,3	1509,4	37,3	12,8	15,7
R104-2A	635,6	11,4	0,1	0,0	850,9	24,7	3,7	3,5	1389,0	31,3	6,7	7,1
R105-2A	761,9	0,6	0,0	0,0	900,4	11,1	1,0	-1,1	1591,6	38,9	13,6	9,9
R106-2A	594,4	2,7	0,0	0,0	912,8	14,5	0,1	0,6	1274,4	30,9	-3,2	-4,7
R107-2A	570,7	7,0	-1,2	0,0	860,9	22,0	5,9	5,1	1282,6	35,9	0,1	1,2
R108-2A	579,6	3,9	0,0	0,0	887,3	18,9	1,0	1,1	-	-	-	-
R109-2A	639,5	0,0	-0,7	0,0	827,6	22,0	5,0	6,2	1410,4	36,9	8,3	9,8
R110-2A	552,2	0,0	0,0	0,0	847,6	18,3	-1,6	-0,8	1301,3	35,1	7,4	4,4
R111-2A	646,3	0,0	-0,7	-1,2	921,5	11,1	0,2	0,7	1386,4	45,6	20,5	20,1
R112-2A	570,8	0,0	0,0	0,0	911,4	14,3	0,4	0,4	1266,6	32,8	-	7,9
R101-2B	773,9	0,0	0,0	0,0	1010,6	23,3	-1,2	4,1	1757,1	40,0	-7,6	7,9
R102-2B	572,7	11,7	0,0	0,0	864,1	19,2	0,2	0,6	1497,3	39,7	5,2	4,3
R103-2B	567,8	2,2	0,0	0,0	814,0	25,1	3,0	1,2	1777,2	45,3	22,0	17,1
R104-2B	656,7	11,8	0,0	0,0	833,4	21,2	2,0	1,3	1361,7	34,0	1,8	0,8
R105-2B	800,0	0,0	-0,7	0,0	906,2	13,8	-0,2	-1,7	1628,4	41,8	13,2	12,9
R106-2B	678,5	0,0	0,0	0,0	901,3	16,3	1,2	1,2	1413,5	35,8	9,7	5,7
R107-2B	591,2	0,0	0,0	0,0	853,1	21,2	4,6	4,8	1489,0	44,3	13,8	6,6
R108-2B	592,6	3,0	0,0	0,0	850,2	19,8	-0,9	-1,2	1262,5	39,4	-	9,0
R109-2B	1075,1	31,0	-0,2	0,0	788,1	18,9	-0,6	-2,0	1465,4	38,3	12,2	9,6
R110-2B	537,8	9,8	0,0	-2,9	868,1	26,9	2,3	1,7	1354,4	37,8	9,4	9,8
R111-2B	666,5	13,6	0,0	1,8	965,2	20,7	3,1	1,9	1234,8	37,4	7,3	8,3
R112-2B	595,2	13,5	0,0	0,0	882,3	20,3	1,7	1,7	1347,9	40,1	-	8,5
R101-2C	680,2	8,2	0,0	-0,6	1013,0	24,5	2,0	2,3	-	-	-	-
R102-2C	597,7	12,9	1,2	1,2	882,7	19,6	1,9	0,3	1939,8	53,8	19,4	25,5
R103-2C	588,0	13,2	0,0	0,0	814,3	24,1	4,4	3,0	1572,3	39,2	8,7	9,0
R104-2C	634,2	26,3	3,0	-3,6	884,6	24,7	7,2	8,4	1408,4	35,7	5,2	6,2
R105-2C	680,5	0,0	0,0	-17,6	933,6	17,8	0,4	-0,4	-	-	-	-
R106-2C	595,1	4,1	0,0	0,0	901,0	19,2	2,2	1,0	1563,9	45,0	13,2	8,4
R107-2C	567,1	5,9	0,0	0,0	829,5	23,4	2,9	-1,2	-	-	-	-
R108-2C	561,2	1,3	0,0	-5,6	910,4	27,8	7,0	7,4	-	-	-	-
R109-2C	684,6	11,7	-0,1	-0,1	822,0	25,8	4,8	3,5	-	-	-	-
R110-2C	558,2	0,0	0,0	0,0	894,1	26,4	5,4	4,4	-	-	-	-
R111-2C	639,6	11,1	-0,4	-0,4	941,2	18,7	1,8	1,8	1283,9	40,1	10,5	14,8
R112-2C	579,2	0,0	0,0	0,0	865,5	21,5	-0,1	-0,1	-	-	-	-
Média		6,7	-0,1	-0,8		20,3	1,9	1,7		34,8	8,6	8,7

definido na Seção 6.2, não obtivemos solução inicial para as R108-2A, R112-2A, R108-2B, R111-2B, R108-2C, R112-2C.

Note que, na Tabela 26, não são obtidas soluções para 9 das 36 instâncias com 40 clientes ao resolver o Modelo M2 no tempo limite de 7200 segundos e, por isso, não são possíveis comparações com as soluções obtidas pelas duas versões da PS. Para estas instâncias, na Tabela 27, apresentamos os valores da função objetivo (VF) das soluções obtidas com a PS1 e a PS2 e a diferença percentual entre eles calculada da seguinte forma:

$$D^1 = \frac{\text{SOL}^{PS1} - \text{SOL}^{PS2}}{\text{SOL}^{PS1}} \times 100$$

onde SOL^{PS1} e SOL^{PS2} são os valores da função objetivo das soluções obtidas com a PS1 e a PS2, respectivamente.

Na Tabela 27, observamos que não são obtidas soluções para três instâncias com a PS1,

Tabela 27 – Comparação entre as soluções obtidas com a PS1 e PS2.

Instância	PS1	PS2	D ¹
	VF	VF	
R102-1A	1389,3	1473,3	-6,0
R108-1A	-	1416,6	-
R101-1B	1741,6	1884,6	-8,2
R105-1C	1555,1	1474,3	5,2
R107-1C	1220,9	1243,6	-1,9
R108-1C	-	1131,1	-
R109-1C	1271,2	1375,8	-8,2
R110-1C	1278,5	1244,5	2,7
R112-1C	-	1211,2	-

pois não é possível encontrar solução inicial para estas instâncias com a estratégia proposta na Seção 6.2. Embora seja possível obter uma solução para todas as instâncias com a PS1, observamos que a PS2 obteve soluções piores para quatro instâncias com diferenças de até 8,2% nos casos em que são obtidas soluções pelas duas versões da PS.

No caso da PS2, avaliamos as melhorias obtidas em relação à solução inicial dada pela heurística construtiva com a qual o método é iniciado. Para isso, reportamos, na Tabela 28, a menor (M^{min}), a maior (M^{max}) e a média ($\bar{M}$) das diferenças percentuais M calculadas da seguinte forma:

$$M = \frac{SOL^{HC} - SOL^{PS2}}{SOL^{HC}} \times 100$$

onde SOL^{HC} é a solução obtida pela heurística construtiva e SOL^{PS2} é a solução obtida pela PS2.

Tabela 28 – Comparação entre a solução inicial (heurística construtiva) com a melhor solução obtida pela PS2.

		M^{min}	M^{max}	$\bar{M}$
Cenário 1	15 clientes	3,4	39,3	16,6
	25 clientes	8,4	60,0	24,6
	40 clientes	9,1	62,1	30,2
		7,0	53,8	23,8
Cenário 2	15 clientes	3,1	32,2	17,6
	25 clientes	7,0	36,8	20,4
	40 clientes	9,6	40,0	22,4
		6,6	36,3	20,1

De acordo com a Tabela 28, tivemos melhorias significativas com diferenças de até 62% no Cenário 1 e de até 40% no Cenário 2. Importante ressaltar que, mesmo no caso em que as soluções dadas pela heurística construtiva são inviáveis por apresentarem tempo extra, as soluções obtidas com a PS2 são todas factíveis, apresentando tempo extra nulo.

6.6 Conclusões

De forma geral, as duas versões da matheurística obtiveram soluções competitivas no caso das instâncias menores, principalmente no Cenário 2 em que o tempo de processamento da matheurística foi menor. Já para as instâncias com 25 e 40 clientes, as soluções obtidas pelas PS1 e PS2 apresentaram soluções com ganhos significativos, principalmente para as instâncias com 40 clientes em que, além de terem ganhos de até 25,5%, o tempo de processamento foi um quarto do tempo gasto pelo *solver* ao resolver o Modelo M2. No caso da PS2, mesmo quando a solução inicial obtida com a heurística construtiva era infactível, obtemos as soluções finais são factíveis para todos as instâncias.

CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, abordamos os problemas de programação do *cross-dock* e de roteamento de veículos de forma integrada para um sistema de distribuição baseado em uma rede de varejo. Para o problema estudado, foram desenvolvidos dois modelos de programação inteira mista, sendo que a diferença entre eles é a inclusão de variáveis indicativas de rotas o que melhorou significativamente o valor das soluções obtidas na relaxação linear. Conforme apontado na revisão da literatura, problemas similares são abordados em [Agustina, Lee e Piplani \(2014\)](#) e, mais recentemente, em [Rahbari et al. \(2019\)](#) para um contexto em que são distribuídos produtos perecíveis. Os problemas apresentam algumas características específicas que não são estendidas para o problema estudado neste trabalho. Diferente dos modelos da literatura, que usam variáveis de três índices para formular as decisões de roteamento, os modelos que desenvolvemos aqui consideram variáveis de dois índices. Com algumas adaptações, mostramos, como esperado, que o desempenho do *solver* ao resolver o modelo com variáveis de dois índices é melhor. Além disso, os modelos desenvolvidos neste trabalho podem ser usados em um contexto de distribuição com produtos intercambiáveis, ou seja, quando determinados produtos podem ser substituídos por outros para satisfazer uma determinada demanda.

De acordo com [Buijs, Vis e Carlo \(2014\)](#), a integração das decisões do *cross-docking* com as decisões da rede da qual ele faz parte pode beneficiar o planejamento do sistema como um todo. Mais especificamente, [Yin, Lyu e Chuang \(2016\)](#) defendem que não faz sentido considerar a programação do *cross-dock* e o problema de roteamento de veículos separadamente quando esses dois problemas fazem parte do sistema e que a integração entre eles contribui para uma distribuição mais eficiente. Nesse sentido, os experimentos computacionais apresentados neste trabalho confirmam a vantagem de resolver os dois problemas de forma integrada.

A estratégia integrada foi comparada com três estratégias diferentes. Uma que considera apenas entregas diretas e outras duas em que a resolução dos problemas é realizada de forma hierárquica; ou seja, em uma, o problema de programação do *cross-dock* é resolvido primeiro e,

depois, o problema de roteamento é tratado em cada uma das docas, e, na outra, essa ordem é invertida. Em geral, verificamos que as soluções obtidas com a estratégia integrada apresentam melhor qualidade com menores custos totais e menos violações dos prazos das entregas. Entre as outras estratégias, a estratégia em que problema de roteamento é resolvido primeiro apresentou as melhores soluções, porém, tiveram soluções com algumas rotas que, quando consideradas no problema de programação *cross-dock*, não eram possíveis de serem completadas antes do tempo limite. Além disso, verificamos que as estratégias hierárquicas não garantem um planejamento ótimo de distribuição para o sistema. Além disso, como esperado, à medida que a dimensão do problema aumenta, constatamos que a estratégia integrada, com a resolução do modelo proposto, não fornece soluções comprovadamente de boa qualidade, ou seja, apresentam *gaps* da ordem de 40% para as instâncias maiores avaliadas do cenário com maior possibilidade de consolidação de cargas. Diante disso, foram desenvolvidos métodos heurísticos para o problema estudado.

Primeiro, desenvolvemos uma heurística construtiva que, como base nos resultados das estratégias hierárquicas, privilegia a construção das rotas e, paralelamente, define a programação do *cross-dock*. A heurística é bastante rápida, porém, seu desempenho foi melhor para o cenário em que as demandas dos clientes são mais baixas, ou seja, é possível ter um número maior de clientes em uma mesma rota.

Na sequência, propomos uma meta-heurística de busca em vizinhança variável. Neste método, definimos algumas vizinhanças baseadas nos diferentes tipos de decisões e as buscas são realizadas com auxílio de um modelo de programação inteira misto adaptado a partir do próprio modelo desenvolvido para o problema. De acordo com resultados obtidos, ao compará-los com as soluções obtidas ao resolver o modelo pelo *solver*, a meta-heurística teve um bom desempenho com soluções competitivas no caso das instâncias menores e, para as instâncias maiores, as soluções foram até 27% melhores com tempo de processamento bem menor (aproximadamente um quarto do tempo de processamento do *solver*).

Por último, adaptamos a matheurística *Proximity Search* para o problema estudado que foi testada com duas maneiras de obter a solução inicial. Uma através do próprio modelo restringindo o espaço de busca com a fixação dos valores de algumas variáveis, e outra com a solução obtida pela heurística construtiva proposta. Nos dois casos, comparamos também com a resolução do modelo diretamente pelo *solver* e, similar aos resultados da meta-heurística, as soluções obtidas foram competitivas para as instâncias menores e melhores para as instâncias maiores com diferenças de até 25,5%. Em especial, ao usarmos a solução inicial obtida com a heurística construtiva, conseguimos obter soluções para todas as instâncias maiores, o que não ocorre para a outra versão da matheurística, para a qual não obtemos soluções para nove instâncias.

Como trabalhos futuros, temos alguns pontos para serem desenvolvidos. Primeiro, do ponto de vista operacional do problema, consideramos interessante investigar o impacto sobre o planejamento ao considerar a ordem de descarregamento dos produtos que estão numa mesma

carga de entrada, já que essa informação poderia ser disponibilizada pelos fornecedores que enviam as cargas. Assim, na prática, a transferência de um determinado produto da doca de recebimento para a doca de saída pode ser iniciada antes de finalizar o processamento da sua carga de entrada. Além disso, a maneira como os produtos podem ser organizados dentro da carga pode ser relevante e, neste caso, decisões sobre o empacotamento dos produtos devem ser consideradas.

Além disso, em relação aos métodos de solução, pretendemos testar outros critérios na heurística construtiva, em especial, para o cenário em que temos menos possibilidades de consolidações para o qual o desempenho da heurística não foi muito bom. Para a meta-heurística, pretendemos incorporar uma fase de diversificação através de um *restart* do método incorporando algumas informações sobre as soluções obtidas na primeira execução. E, também, tentar incorporar a matheurística *Proximity Search* como uma fase da meta-heurística. Tendo em vista o uso bem sucedido de estratégias de decomposição para problemas integrados, acreditamos que essa possa ser uma direção promissora para pesquisas futuras. Também consideramos, como passo futuro, buscar instâncias reais para validação das estratégias propostas.

REFERÊNCIAS

AGUSTINA, D.; LEE, C.; PIPLANI, R. A review: Mathematical modles for cross docking planning. **International Journal of Engineering Business Management**, v. 2, p. 47–54, 2010. Citado na página 28.

_____. Vehicle scheduling and routing at a cross docking center for food supply chains. **International Journal of Production Economics**, v. 152, p. 29–41, 2014. Citado nas páginas 24, 32, 34, 41, 44, 45, 46, 49, 52, 59, 67, 68, 69, 76 e 119.

ÁLVAREZ-MIRANDA, E.; CACCHIANI, V.; LODI, A.; PARRIANI, T.; SCHMIDT, D. R. Single-commodity robust network design problem: Complexity, instances and heuristic solutions. **European Journal of Operational Research**, v. 238, p. 711–723, 2014. Citado na página 110.

ALVAREZ-PEREZ, G.; GONZÁLEZ-VELARDE, J.; FOWLER, J. W. Crossdocking—just in time scheduling: an alternative solution approach. **Journal of the Operational Research Society**, v. 60, p. 554–564, 2009. Citado nas páginas 32, 34, 35 e 38.

ARABANI, A. B.; GHOMI, S. F.; ZANDIEH, M. A multi-criteria cross-docking scheduling with just-in-time approach. **The International Journal of Advanced Manufacturing Technology**, v. 49, p. 741–756, 2010. Citado nas páginas 31 e 32.

_____. Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. **Expert systems with Applications**, v. 38, p. 1964–1979, 2011. Citado nas páginas 31, 32, 37 e 40.

ASSADI, M. T.; BAGHERI, M. Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. **Computers & Industrial Engineering**, v. 96, p. 149–161, 2016. Citado nas páginas 32, 34, 38, 39 e 46.

BELLE, J. V.; VALCKENAERS, P.; BERGHE, G. V.; CATTRYSSSE, D. A tabu search approach to the truck scheduling problem with multiple docks and time windows. **Computers & Industrial Engineering**, v. 66, p. 818–826, 2013. Citado nas páginas 33, 34, 38 e 46.

BELLE, J. V.; VALCKENAERS, P.; CATTRYSSSE, D. Cross-docking: State of the art. **Omega**, v. 40, p. 827–846, 2012. Citado nas páginas 28, 29, 30 e 46.

BERGHMAN, L.; BRIAND, C.; LEUS, R.; LOPEZ, P. The truck scheduling problem at crossdocking terminals-exclusive versus mixed mode. In: **4th International Conference on Operations Research and Enterprise Systems (ICORES 2015)**. [S.l.: s.n.], 2015. p. pp–247. Citado nas páginas 40 e 46.

BERNARDES, E.; ARMENTANO, V. A.; TOLEDO, F. Programação do cross-docking integrada às decisões de roteamento no processo de distribuição. In: **Anais do XLIX Simpósio Brasileiro de Pesquisa Operacional**. [s.n.], 2017. p. 1691–1702. Disponível em: <<http://www.sbpo2017.iltc.br/pdf/168858.pdf>>. Citado na página 47.

- BERTHOLD, T.; LODI, A.; SALVAGNIN, D. Ten years of feasibility pump, and counting. **EURO Journal on Computational Optimization**, v. 7, p. 1–14, 2019. Citado na página 109.
- BOWMAN, E. H. The schedule-sequencing problem. **Operations Research**, v. 7, p. 621–624, 1959. Citado nas páginas 35, 36, 39 e 46.
- BOYSEN, N. Truck scheduling at zero-inventory cross docking terminals. **Computers & Operations Research**, v. 37, p. 32–41, 2010. Citado nas páginas 24, 27, 29, 30, 32 e 34.
- BOYSEN, N.; BRISKORN, D.; TSCHÖKE, M. Truck scheduling in cross-docking terminals with fixed outbound departures. **OR spectrum**, v. 35, p. 479–504, 2013. Citado nas páginas 33 e 34.
- BOYSEN, N.; FLIEDNER, M. Cross dock scheduling: Classification, literature review and research agenda. **Omega**, v. 38, p. 413–422, 2010. Citado na página 24.
- BUIJS, P.; VIS, I. F.; CARLO, H. J. Synchronization in cross-docking networks: A research classification and framework. **European Journal of Operational Research**, v. 239, p. 593–608, 2014. Citado nas páginas 24, 28, 45 e 119.
- CHEN, F.; LEE, C.-Y. Minimizing the makespan in a two-machine cross-docking flow shop problem. **European Journal of Operational Research**, v. 193, p. 59–72, 2009. Citado na página 35.
- CHEN, F.; SONG, K. Minimizing makespan in two-stage hybrid cross docking scheduling problem. **Computers & Operations Research**, v. 36, p. 2066–2073, 2009. Citado nas páginas 32, 34, 38 e 46.
- CHEN, H. Fix-and-optimize and variable neighborhood search approaches for multi-level capacitated lot sizing problems. **Omega**, v. 56, p. 25–36, 2015. Citado na página 89.
- DONDO, R.; CERDÁ, J. A sweep-heuristic based formulation for the vehicle routing problem with cross-docking. **Computers & Chemical Engineering**, v. 48, p. 293–311, 2013. Citado na página 43.
- _____. A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors. **Computers & Chemical Engineering**, v. 63, p. 184–205, 2014. Citado na página 43.
- _____. The heterogeneous vehicle routing and truck scheduling problem in a multi-door cross-dock system. **Computers & Chemical Engineering**, v. 76, p. 42–62, 2015. Citado na página 43.
- DONDO, R.; MÉNDEZ, C. A.; CERDÁ, J. Managing distribution in supply chain networks. **Industrial & Engineering Chemistry Research**, v. 48, p. 9961–9978, 2009. Citado na página 43.
- _____. The multi-echelon vehicle routing problem with cross docking in supply chain management. **Computers & Chemical Engineering**, v. 35, p. 3002–3024, 2011. Citado na página 43.
- FISCHETTI, M.; FISCHETTI, M.; MONACI, M. Optimal turbine allocation for offshore and onshore wind farms. In: **Optimization in the Real World**. [S.l.: s.n.], 2016. p. 55–78. Citado na página 109.

FISCHETTI, M.; MONACI, M. Proximity search for 0-1 mixed-integer convex programming. **Journal of Heuristics**, v. 20, p. 709–731, 2014. Citado na página 109.

_____. Proximity search heuristics for wind farm optimal layout. **Journal of Heuristics**, v. 22, p. 459–474, 2016. Citado na página 109.

FOROUHARFARD, S.; ZANDIEH, M. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. **The International Journal of Advanced Manufacturing Technology**, v. 51, p. 1179–1193, 2010. Citado nas páginas 31 e 32.

GHIANI, G.; LAPORTE, G.; MUSMANNO, R. **Introduction to logistics systems planning and control**. [S.l.]: John Wiley & Sons, 2004. Citado na página 23.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search: Principles and applications. **European journal of operational research**, v. 130, p. 449–467, 2001. Citado nas páginas 89 e 90.

JOO, C. M.; KIM, B. S. Scheduling compound trucks in multi-door cross-docking terminals. **The International Journal of Advanced Manufacturing Technology**, v. 64, p. 977–988, 2013. Citado nas páginas 34, 39 e 46.

KESHTZARI, M.; NADERI, B.; MEHDIZADEH, E. An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems. **Computers & Industrial Engineering**, v. 91, p. 197–204, 2016. Citado nas páginas 31, 32, 40 e 46.

LADIER, A.-L.; ALPAN, G. Crossdock truck scheduling with time windows: Earliness, tardiness and storage policies. **Journal of Intelligent Manufacturing**, p. 1–15, 2014. Citado na página 35.

_____. Integrating truck scheduling and employee rostering in a cross-docking platform—an iterative approach. In: **Industrial Engineering and Systems Management (IESM), 2015 International Conference on**. [S.l.: s.n.], 2015. p. 676–685. Citado nas páginas 34, 40 e 41.

_____. Cross-docking operations: Current research versus industry practice. **Omega**, v. 62, p. 145–162, 2016. Citado nas páginas 17, 29, 30, 31, 34, 35, 45 e 46.

_____. Crossdock truck scheduling with time windows: earliness, tardiness and storage policies. **Journal of Intelligent Manufacturing**, v. 29, p. 569–583, 2018. Citado nas páginas 32 e 34.

LEE, Y. H.; JUNG, J. W.; LEE, K. M. Vehicle routing scheduling for cross-docking in the supply chain. **Computers & Industrial Engineering**, v. 51, p. 247–256, 2006. Citado nas páginas 41 e 42.

LI, L.; SONG, S.; WU, C.; WANG, R. Fix-and-optimize and variable neighborhood search approaches for stochastic multi-item capacitated lot-sizing problems. **Mathematical Problems in Engineering**, v. 2017, 2017. Citado na página 89.

LI, Y.; LIM, A.; RODRIGUES, B. Crossdocking—jit scheduling with time windows. **Journal of the Operational Research Society**, v. 55, p. 1342–1351, 2004. Citado nas páginas 32, 34, 37, 38, 39 e 46.

LIAO, C.-J.; LIN, Y.; SHIH, S. C. Vehicle routing with cross-docking in the supply chain. **Expert Systems with Applications**, v. 37, p. 6868–6873, 2010. Citado na página 42.

- LIAO, T.; EGBELU, P.; CHANG, P.-C. Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations. **International Journal of Production Economics**, v. 141, p. 212–229, 2013. Citado nas páginas 33 e 34.
- LIAO, T. W.; EGBELU, P. J.; CHANG, P.-C. Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. **Applied Soft Computing**, v. 12, p. 3683–3697, 2012. Citado nas páginas 31 e 32.
- MAKNOON, Y.; LAPORTE, G. Vehicle routing with cross-dock selection. **Computers & Operations Research**, v. 77, p. 254–266, 2017. Citado na página 43.
- MANNE, A. S. On the job-shop scheduling problem. **Operations Research**, v. 8, p. 219–223, 1960. Citado nas páginas 36 e 46.
- MCWILLIAMS, D. L. Iterative improvement to solve the parcel hub scheduling problem. **Computers & Industrial Engineering**, v. 59, p. 136–144, 2010. Citado nas páginas 33 e 34.
- MCWILLIAMS, D. L.; MCBRIDE, M. E. A beam search heuristics to solve the parcel hub scheduling problem. **Computers & Industrial Engineering**, v. 62, p. 1080–1092, 2012. Citado nas páginas 33 e 34.
- MCWILLIAMS, D. L.; STANFIELD, P. M.; GEIGER, C. D. The parcel hub scheduling problem: A simulation-based solution approach. **Computers & Industrial Engineering**, v. 49, p. 393–412, 2005. Citado nas páginas 33 e 34.
- _____. Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers. **Computers & Industrial Engineering**, v. 54, p. 709–720, 2008. Citado nas páginas 33 e 34.
- MLADENOVIC, N. A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization. In: **papers presented at Optimization Days**. [S.l.: s.n.], 1995. v. 12. Citado na página 89.
- MLADENović, N.; HANSEN, P. Variable neighborhood search. **Computers & operations research**, v. 24, p. 1097–1100, 1997. Citado na página 89.
- MOGHADAM, S. S.; GHOMI, S. F.; KARIMI, B. Vehicle routing scheduling problem with cross docking and split deliveries. **Computers & Chemical Engineering**, v. 69, p. 98–107, 2014. Citado na página 42.
- MOHTASHAMI, A. Scheduling trucks in cross docking systems with temporary storage and repetitive pattern for shipping trucks. **Applied Soft Computing**, v. 36, p. 468–486, 2015. Citado nas páginas 24 e 27.
- MORAIS, V. W.; MATEUS, G. R.; NORONHA, T. F. Iterated local search heuristics for the vehicle routing problem with cross-docking. **Expert Systems with Applications**, v. 41, p. 7495–7506, 2014. Citado na página 42.
- NIKOLOPOULOU, A. I.; REPOUSSIS, P. P.; TARANTILIS, C. D.; ZACHARIADIS, E. E. Moving products between location pairs: Cross-docking versus direct-shipping. **European Journal of Operational Research**, v. 256, p. 803–819, 2017. Citado na página 43.

- PAN, C.-H. A study of integer programming formulations for scheduling problems. **International Journal of Systems Science**, v. 28, p. 33–41, 1997. Citado na página [35](#).
- RAHBARI, A.; NASIRI, M. M.; WERNER, F.; MUSAVI, M.; JOLAI, F. The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. **Applied Mathematical Modelling**, v. 70, p. 605–625, 2019. Citado nas páginas [24](#), [32](#), [34](#), [44](#), [45](#), [46](#), [49](#), [52](#), [59](#), [67](#), [69](#) e [119](#).
- ROSALES, C. R.; FRY, M. J.; RADHAKRISHNAN, R. Transfreight reduces costs and balances workload at georgetown crossdock. **Interfaces**, v. 39, p. 316–328, 2009. Citado nas páginas [33](#) e [34](#).
- SANTOS, F. A.; MATEUS, G. R.; CUNHA, A. S. D. The pickup and delivery problem with cross-docking. **Computers & Operations Research**, v. 40, p. 1085–1093, 2013. Citado na página [42](#).
- SHAKERI, M.; LOW, M. Y. H.; TURNER, S. J.; LEE, E. W. A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. **Computers & Operations Research**, v. 39, p. 2564–2577, 2012. Citado na página [35](#).
- SHIGUEMOTO, A. L.; NETTO, U. S. C.; BAUAB, G. H. S. An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. **International Journal of Production Research**, Taylor & Francis, v. 52, p. 1231–1239, 2014. Citado nas páginas [31](#) e [32](#).
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations research**, v. 35, p. 254–265, 1987. Citado nas páginas [62](#), [68](#) e [144](#).
- SOLTANI, R.; SADJADI, S. J. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. **Transportation Research Part E: Logistics and Transportation Review**, v. 46, p. 650–666, 2010. Citado nas páginas [31](#) e [32](#).
- TARANTILIS, C. D. Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. **Optimization letters**, v. 7, p. 1583–1596, 2013. Citado na página [42](#).
- TOSCHI, M.; LANZARONE, E.; ANAYA-ARENAS, A. M.; BÉLANGER, V.; NICOLETTA, V.; RUIZ, A. A fix-and-optimize variable neighborhood search for the biomedical sample transportation problem. **IFAC-PapersOnLine**, v. 51, p. 992–997, 2018. Citado na página [89](#).
- VAHDANI, B.; SOLTANI, R.; ZANDIEH, M. Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. **The International Journal of Advanced Manufacturing Technology**, v. 46, p. 769–783, 2010. Citado nas páginas [31](#) e [32](#).
- VAHDANI, B.; TAVAKKOLI-MOGHADDAM, R.; ZANDIEH, M.; RAZMI, J. Vehicle routing scheduling using an enhanced hybrid optimization approach. **Journal of Intelligent Manufacturing**, v. 23, p. 759–774, 2012. Citado na página [42](#).
- VAHDANI, B.; ZANDIEH, M. Scheduling trucks in cross-docking systems: Robust meta-heuristics. **Computers & Industrial Engineering**, v. 58, p. 12–24, 2010. Citado nas páginas [37](#) e [40](#).
- VOGT, J. J. The successful cross-dock based supply chain. **Journal of Business Logistics**, v. 31, p. 99–119, 2010. Citado na página [27](#).

WAGNER, H. M. An integer linear-programming model for machine scheduling. **Naval Research Logistics (NRL)**, v. 6, p. 131–140, 1959. Citado nas páginas 35, 36, 39 e 46.

WEN, M.; LARSEN, J.; CLAUSEN, J.; CORDEAU, J.-F.; LAPORTE, G. Vehicle routing with cross-docking. **Journal of the Operational Research Society**, v. 60, p. 1708–1718, 2009. Citado na página 42.

YIN, P.-Y.; LYU, S.-R.; CHUANG, Y.-L. Cooperative coevolutionary approach for integrated vehicle routing and scheduling using cross-dock buffering. **Engineering Applications of Artificial Intelligence**, v. 52, p. 40–53, 2016. Citado nas páginas 24, 43, 45 e 119.

YU, H.; THAPA, G. B. A cross-docking framework for supply chain logistics in production scheduling. **Journal of Institute of Science and Technology**, v. 19, p. 39–44, 2015. Citado nas páginas 39 e 46.

YU, W.; EGBELU, P. J. Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. **European Journal of Operational Research**, v. 184, p. 377–396, 2008. Citado nas páginas 31, 32, 34, 36, 37, 40 e 46.

DETALHES SOBRE AS SOLUÇÕES OBTIDAS NO CAPÍTULO 3

Neste apêndice, são reportados mais detalhes sobre as soluções obtidas nos experimentos computacionais que foram utilizadas para as análises apresentadas no Capítulo 3.

A seguir, na Seção A.1, temos os valores obtidos para a função objetivo com a relaxação linear utilizando os dois modelos. Na Seção A.2, apresentamos informações sobre as soluções obtidas com os dois modelos desenvolvidos que são usadas na comparação entre eles. Na sequência, com o Modelo M2, apresentamos na Seção A.3 as soluções obtidas com cada uma das estratégias propostas para obter um plano de distribuição. Por último, na Seção A.4, apresentamos um gráfico de comparação entre as estratégias EI e RCD para o conjunto de instâncias de 15 clientes do Cenário 1, similar ao gráfico apresentado no Capítulo 3 com as instâncias do Cenário 2.

Nas próximas seções, todas as tabelas estão divididas em três partes que se referem, de cima para baixo, às instâncias dos Grupos A, B e C, nessa ordem. Na última linha de cada tabela, são apresentados a média dos valores de cada coluna.

A.1 Soluções obtidas com a relaxação linear dos dois modelos

Nas Tabelas 29 e 30, respectivamente, para o Cenário 1 e para o Cenário 2, temos os valores da função objetivo obtidos com a relaxação linear utilizando os Modelos M1 e M2 com o limitante inferior (LI) e sem para as variáveis que indicam o fim da operação de carregamento de cada carga de saída (dt_i).

Notamos um aumento significativo dos valores para todas instâncias com a inclusão das restrições (3.35), que impõem limitante inferior para a variável dt_i . Além disso, os valores

obtidos com a relaxação linear do Modelo M2 são maiores comparados aos valores obtidos com o Modelo M1 em todos os casos, independentemente da inclusão ou não das restrições (3.35).

Tabela 29 – Soluções obtidas com a relaxação linear dos Modelos M1 e M2 - Cenário 1.

Instância	15 clientes				25 clientes			
	Sem LI		Com LI		Sem LI		Com LI	
	M1	M2	M1	M2	M1	M2	M1	M2
R101-1A	234,9	348,7	536,2	646,8	335,0	702,6	726,4	1086,6
R102-1A	298,5	486,5	645,3	827,7	382,6	844,2	725,2	1177,0
R103-1A	272,7	463,5	630,2	815,6	352,0	797,9	749,8	1186,8
R104-1A	251,1	408,4	572,1	724,2	353,1	767,7	746,6	1151,8
R105-1A	275,3	465,8	622,7	807,8	357,9	810,1	751,2	1194,4
R106-1A	262,7	429,9	616,8	777,8	345,7	752,7	678,0	1075,6
R107-1A	270,6	450,7	658,4	833,0	378,5	822,4	751,1	1186,0
R108-1A	238,2	365,8	510,6	634,3	385,8	826,5	810,8	1241,1
R109-1A	243,9	383,3	531,4	667,1	327,6	674,3	612,9	952,0
R110-1A	258,8	440,1	656,1	830,4	358,6	810,6	728,1	1170,3
R111-1A	261,5	450,9	626,3	810,2	335,8	752,6	617,9	1025,1
R112-1A	261,9	434,0	573,3	739,9	366,5	803,9	750,9	1179,3
R101-1B	234,7	348,6	448,2	558,9	335,0	702,6	616,0	976,3
R102-1B	298,5	486,3	714,6	897,0	382,6	844,6	784,9	1236,7
R103-1B	272,7	463,4	613,1	798,5	352,0	797,9	700,8	1137,8
R104-1B	249,9	406,8	507,8	661,0	353,1	767,7	664,7	1069,8
R105-1B	275,3	465,8	583,9	769,0	357,9	810,1	732,6	1175,8
R106-1B	262,7	429,4	542,5	703,5	345,7	752,7	668,9	1066,6
R107-1B	270,6	450,7	650,7	825,3	378,5	822,4	725,0	1159,8
R108-1B	237,4	365,7	489,1	613,6	385,8	826,5	750,6	1180,9
R109-1B	242,9	383,3	480,3	617,1	327,6	674,3	571,8	910,9
R110-1B	258,4	438,5	632,0	807,0	358,6	810,0	667,9	1110,1
R111-1B	260,9	450,8	548,9	733,4	335,8	752,6	789,8	1197,3
R112-1B	261,9	434,0	661,2	827,9	366,5	803,9	666,1	1094,5
R101-1C	234,7	348,7	434,9	545,6	335,0	702,6	561,5	921,8
R102-1C	298,5	486,2	632,6	815,0	382,6	844,2	669,4	1121,2
R103-1C	272,7	463,5	558,0	743,4	352,0	798,0	774,8	1211,8
R104-1C	249,9	406,7	536,9	690,2	353,1	767,8	774,7	1179,9
R105-1C	275,3	465,7	586,9	771,9	357,9	810,2	638,7	1081,9
R106-1C	262,7	429,1	543,0	703,9	345,7	752,8	728,9	1126,6
R107-1C	270,6	450,7	630,3	804,9	378,5	822,5	738,5	1173,3
R108-1C	237,4	365,7	498,1	622,7	385,8	825,5	715,1	1145,5
R109-1C	242,9	383,3	470,7	607,5	327,6	674,3	580,9	920,0
R110-1C	258,4	438,4	586,4	761,1	358,6	810,0	609,3	1051,5
R111-1C	260,9	450,8	630,8	815,3	335,8	752,6	588,1	995,3
R112-1C	261,9	434,0	503,3	669,9	366,5	804,0	635,3	1063,8
Média	260,6	427,0	574,0	735,5	356,6	780,4	694,5	1109,3

Fonte: Elaborada pelo autor.

Tabela 30 – Soluções obtidas com a relaxação linear dos Modelos M1 e M2 - Cenário 2.

Instância	15 clientes				25 clientes			
	Sem LI		Com LI		Sem LI		Com LI	
	M1	M2	M1	M2	M1	M2	M1	M2
R101-2A	223,6	283,7	400,2	459,8	304,9	461,7	467,1	620,8
R102-2A	223,8	277,7	350,5	404,2	303,7	433,9	465,9	593,1
R103-2A	222,2	283,1	404,0	464,6	302,7	423,8	435,1	553,1
R104-2A	223,7	277,6	406,1	459,8	303,1	431,5	439,5	564,8
R105-2A	222,0	292,1	424,1	493,6	303,7	471,2	496,7	661,0
R106-2A	222,1	292,0	386,5	455,8	302,4	461,2	520,0	675,7
R107-2A	223,3	280,4	374,2	431,1	303,2	435,5	462,2	591,6
R108-2A	223,4	283,0	376,4	435,8	302,6	453,1	514,5	662,1
R109-2A	224,8	283,2	412,9	471,0	300,9	421,7	456,6	574,4
R110-2A	222,8	276,2	396,3	449,5	302,3	449,2	488,3	632,2
R111-2A	224,7	301,1	432,8	508,6	304,8	492,3	504,5	687,9
R112-2A	224,7	292,3	397,6	464,6	305,8	465,1	553,7	710,0
R101-2B	219,2	283,6	409,2	473,2	303,7	461,7	446,2	601,1
R102-2B	222,1	277,7	343,4	398,8	301,0	433,9	446,3	576,1
R103-2B	219,4	283,1	370,0	433,4	300,2	423,8	417,9	538,5
R104-2B	223,7	277,6	421,9	475,6	302,2	431,5	454,7	580,9
R105-2B	221,3	292,1	431,5	501,7	302,9	471,2	479,2	644,3
R106-2B	221,6	292,0	411,3	481,1	302,7	461,2	498,6	654,0
R107-2B	222,5	280,4	406,9	464,6	303,1	435,4	475,5	605,0
R108-2B	221,7	282,9	395,9	456,9	302,5	453,0	471,8	619,5
R109-2B	221,3	283,1	472,8	534,3	300,3	421,7	457,3	575,7
R110-2B	219,7	276,2	351,0	407,4	302,3	449,2	439,6	583,5
R111-2B	221,6	301,1	381,8	460,7	305,4	492,3	464,6	647,4
R112-2B	223,4	292,2	345,6	413,9	304,9	465,1	487,8	645,0
R101-2C	219,2	283,7	365,8	429,8	303,7	461,7	439,1	594,1
R102-2C	220,9	277,7	358,9	415,6	301,0	433,9	461,9	591,7
R103-2C	218,0	283,1	350,3	415,1	300,1	423,9	439,0	559,7
R104-2C	218,9	277,6	328,9	387,4	302,2	431,5	460,5	586,8
R105-2C	221,2	292,1	397,8	468,1	302,8	471,2	455,2	620,5
R106-2C	221,6	292,0	378,0	447,9	302,4	461,2	473,0	628,7
R107-2C	220,2	280,4	366,0	426,0	303,1	435,5	420,4	549,9
R108-2C	219,6	283,0	376,1	439,3	302,5	453,1	442,5	590,2
R109-2C	219,4	283,2	408,6	472,0	300,3	421,8	428,2	546,6
R110-2C	219,7	276,2	391,0	447,4	302,3	449,2	463,3	607,2
R111-2C	221,6	301,1	361,2	440,2	302,8	492,3	462,8	648,2
R112-2C	223,4	292,3	390,2	458,5	304,9	465,2	458,9	616,1
Média	221,7	285,2	388,2	451,3	302,8	450,0	465,2	609,4

Fonte: Elaborada pelo autor.

A.2 Soluções obtidas com os dois modelos

Nas Tabelas 31 e 32, respectivamente para o Cenário 1 e para o Cenário 2, apresentamos informações sobre as soluções obtidas pelo *solver* ao resolver os Modelos M1 e M2 para cada instância. O tempo de solução é limitado a 7200 segundos. Nas tabelas, são reportados o valor da função objetivo (VF) e o *gap* de otimalidade (GAP) em porcentagem das soluções e o tempo em segundos (T) para obtê-las.

Tabela 31 – Informações sobre as soluções obtidas com os dois modelos desenvolvido - Cenário 1.

Instância	15 clientes						25 clientes					
	Modelo M1			Modelo M2			Modelo M1			Modelo M2		
	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)
R101-1A	1090,0	0,0	33	1090,0	0,0	24	1532,1	0,0	1711	1532,1	0,0	1214
R102-1A	1089,8	0,0	12	1089,8	0,0	2	1499,1	0,7	7200	1498,6	0,4	7200
R103-1A	999,3	0,0	104	999,3	0,0	8	1485,1	2,3	7200	1486,6	2,0	7200
R104-1A	905,4	0,0	12	905,4	0,0	3	1395,7	2,6	7200	1390,7	2,3	7200
R105-1A	1119,3	0,0	2629	1119,3	0,0	890	1545,8	3,7	7200	1545,6	3,4	7200
R106-1A	982,2	0,0	10	982,2	0,0	3	1421,1	6,0	7200	1421,1	4,9	7200
R107-1A	1062,9	0,0	9	1062,9	0,0	3	1470,9	0,7	7200	1470,9	0,5	7200
R108-1A	788,9	0,0	197	788,9	0,0	169	1478,5	2,5	7200	1475,2	2,3	7200
R109-1A	855,5	0,0	538	855,5	0,0	173	1286,1	6,1	7200	1277,4	4,0	7200
R110-1A	1058,4	0,0	3	1058,4	0,0	1	1439,6	0,9	7200	1438,4	0,9	7200
R111-1A	1042,6	0,0	542	1042,6	0,0	330	1394,6	6,1	7200	1393,0	2,4	7200
R112-1A	943,2	0,0	1852	943,2	0,0	674	1533,6	3,1	7200	1530,5	2,9	7200
R101-1B	914,2	3,8	7200	914,2	2,7	7200	1519,3	15,7	7200	1498,4	13,7	7200
R102-1B	1173,7	0,0	10	1173,7	0,0	4	1530,8	4,3	7200	1525,1	3,9	7200
R103-1B	994,6	0,0	353	994,6	0,0	75	1480,4	4,9	7200	1481,4	3,1	7200
R104-1B	880,4	0,0	64	880,4	0,0	17	1411,4	8,5	7200	1425,0	9,4	7200
R105-1B	1048,4	0,0	11	1048,4	0,0	4	1533,3	5,4	7200	1540,3	5,1	7200
R106-1B	982,6	0,0	17	982,6	0,0	6	1437,0	8,0	7200	1430,2	5,5	7200
R107-1B	1024,2	0,0	278	1024,2	0,0	91	1475,8	4,5	7200	1477,7	4,0	7200
R108-1B	806,8	0,0	1323	806,8	0,0	691	1473,4	6,2	7200	1465,7	4,4	7200
R109-1B	896,7	2,8	7200	896,7	1,1	7200	1312,3	11,7	7200	1329,2	12,3	7200
R110-1B	1042,2	1,8	7200	1042,2	0,0	915	1447,6	5,9	7200	1447,6	3,9	7200
R111-1B	1066,4	0,0	32	1066,4	0,0	22	1749,7	6,2	7200	1749,7	5,6	7200
R112-1B	1032,8	0,0	499	1032,8	0,0	68	1543,4	4,7	7200	1542,6	4,3	7200
R101-1C	1040,8	12,3	7200	1031,1	10,2	7200	1713,5	27,6	7200	1676,9	23,9	7200
R102-1C	1154,4	0,0	3797	1154,4	0,0	3593	1577,6	13,6	7200	1569,2	9,4	7200
R103-1C	1005,9	0,0	113	1005,9	0,0	42	1688,6	8,0	7200	1662,7	6,3	7200
R104-1C	878,9	0,0	2915	878,9	0,0	899	1436,5	2,6	7200	1436,5	2,5	7200
R105-1C	1365,9	12,7	7200	1360,0	11,7	7200	1619,4	16,0	7200	1636,9	13,7	7200
R106-1C	1173,7	0,0	2728	1173,7	0,0	534	1478,1	6,8	7200	1465,0	5,5	7200
R107-1C	1019,8	0,0	909	1019,8	0,0	175	1497,3	7,2	7200	1491,5	5,7	7200
R108-1C	823,5	2,3	7200	823,5	0,0	4958	1480,2	3,3	7200	1480,2	3,1	7200
R109-1C	915,1	0,0	93	915,1	0,0	72	1341,2	12,7	7200	1298,2	9,5	7200
R110-1C	982,4	1,2	7200	982,4	0,0	4401	1559,3	19,0	7200	1511,3	13,0	7200
R111-1C	1035,8	2,6	7200	1035,8	1,4	7200	1442,8	9,5	7200	1439,4	8,2	7200
R112-1C	998,0	2,7	7200	998,0	0,0	4575	1574,2	12,7	7200	1564,8	11,6	7200
Média	1005,4	1,2	2330	1005,0	0,8	1651	1494,6	7,2	7048	1489,0	5,9	7034

Fonte: Elaborada pelo autor.

No caso do *solver* de otimização Gurobi, o *gap* de otimalidade fornecido é calculado da seguinte forma:

$$\text{GAP} = \frac{\text{LP} - \text{LD}}{\text{LP}} \times 100,$$

onde LD e LP são os limitante primal e dual respectivamente.

É possível observar que, no Cenário 1 e no Cenário 2, o Modelo M2 apresenta, em média, soluções com *gaps* e tempos menores para os dois conjuntos de instâncias, principalmente, para as instâncias do Cenário 1. Diante disso, o Modelo M2 foi escolhido para ser usado nos testes em que avaliamos a relevância de integrar os problemas de programação de *cross-dock* e de roteamento.

Tabela 32 – Informações sobre as soluções obtidas com os dois modelos desenvolvido - Cenário 2.

Instância	15 clientes						25 clientes					
	Modelo M1			Modelo M2			Modelo M1			Modelo M2		
	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)
R101-2A	682,4	0,0	324	682,4	0,0	105	977,1	17,0	7200	955,2	15,5	7200
R102-2A	612,9	17,6	7200	610,5	15,4	7200	877,4	22,8	7200	843,6	14,0	7200
R103-2A	633,4	0,0	2747	633,4	0,0	3324	774,5	27,5	7200	797,1	21,7	7200
R104-2A	634,7	9,4	7200	634,7	6,8	7200	831,0	31,8	7200	828,2	21,4	7200
R105-2A	761,9	0,0	1649	761,9	0,0	1821	931,9	12,9	7200	900,4	9,1	7200
R106-2A	594,4	0,0	1227	594,4	0,0	2300	922,4	18,3	7200	911,2	12,2	7200
R107-2A	570,7	4,5	7200	570,7	3,9	7200	818,9	22,0	7200	839,3	18,2	7200
R108-2A	579,6	0,0	3796	579,6	0,0	3571	890,3	22,9	7200	878,1	16,6	7200
R109-2A	639,5	0,0	2388	639,5	0,0	1069	782,6	22,3	7200	767,2	15,6	7200
R110-2A	552,2	0,0	733	552,2	0,0	1207	853,5	21,8	7200	847,6	16,2	7200
R111-2A	646,3	0,0	1181	646,3	0,0	800	926,1	14,3	7200	921,5	9,4	7200
R112-2A	570,8	0,0	1142	570,8	0,0	1199	910,6	14,8	7200	911,4	11,0	7200
R101-2B	773,9	0,0	1394	773,9	0,0	1182	980,1	20,9	7200	987,3	20,4	7200
R102-2B	572,7	7,9	7200	572,7	4,3	7200	861,6	23,2	7200	854,5	16,5	7200
R103-2B	567,8	0,0	875	567,8	0,0	1997	810,1	32,6	7200	807,9	24,0	7200
R104-2B	656,7	8,8	7200	656,7	7,6	7200	810,8	28,5	7200	833,4	20,6	7200
R105-2B	800,0	0,0	1849	800,0	0,0	1531	906,3	10,8	7200	903,0	11,2	7200
R106-2B	678,5	0,0	4237	678,5	0,0	1554	885,4	19,6	7200	899,6	14,7	7200
R107-2B	591,2	0,0	1064	591,2	0,0	779	838,9	23,2	7200	828,9	17,0	7200
R108-2B	592,6	0,0	2670	592,6	0,0	2455	860,6	25,1	7200	849,3	18,3	7200
R109-2B	1078,4	31,0	7200	1075,1	24,3	7200	796,7	24,5	7200	788,1	18,1	7200
R110-2B	537,8	6,1	7200	537,8	6,5	7200	853,0	28,3	7200	867,8	25,9	7200
R111-2B	654,4	9,2	7200	654,4	9,5	7200	928,4	19,6	7200	921,2	15,3	7200
R112-2B	595,6	13,5	7200	595,2	4,5	7200	870,8	19,7	7200	870,5	16,4	7200
R101-2C	680,2	0,0	2311	680,2	0,0	4370	1119,5	32,4	7200	994,5	22,2	7200
R102-2C	590,6	6,8	7200	597,7	9,3	7200	876,3	24,0	7200	882,7	19,2	7200
R103-2C	588,0	11,3	7200	588,0	9,2	7200	789,9	28,2	7200	807,8	23,2	7200
R104-2C	619,2	27,9	7200	615,0	20,4	7200	808,5	27,5	7200	830,9	19,7	7200
R105-2C	680,5	0,0	449	680,5	0,0	776	984,4	25,5	7200	931,9	0,0	3477
R106-2C	595,1	0,0	7057	595,1	0,0	3385	909,7	25,6	7200	887,7	17,3	7200
R107-2C	567,1	0,0	6968	567,1	1,8	7200	849,3	31,6	7200	809,0	21,1	7200
R108-2C	561,2	0,0	2069	561,2	0,0	1890	846,6	26,9	7200	859,9	23,0	7200
R109-2C	688,0	9,6	7200	684,6	6,7	7200	805,7	31,3	7200	786,9	21,9	7200
R110-2C	558,2	0,0	4790	558,2	0,0	968	850,2	27,5	7200	863,2	22,8	7200
R111-2C	639,6	6,5	7200	637,3	3,6	7200	941,3	21,4	7200	931,8	16,8	7200
R112-2C	579,2	0,0	1722	579,2	0,0	1083	895,3	25,2	7200	865,5	21,0	7200
Média	636,8	4,7	4262,3	636,6	3,7	4037,9	877,1	23,7	7200,0	868,4	17,4	7096,6

Fonte: Elaborada pelo autor.

A.3 Soluções obtidas com as estratégias propostas

Nesta seção, apresentamos as soluções obtidas com cada uma das estratégias propostas para obter um plano de distribuição para o sistema: estratégia integrada (EI), estratégia roteamento/*cross-dock* (RCD), estratégia *cross-dock*/roteamento (CDR) e estratégia de entrega direta (DD). Nas comparações entre as estratégias apresentadas no Capítulo 3, as análises são feitas com base nos valores das soluções e informações sobre elas apresentados aqui.

A.3.1 Estratégia integrada

As soluções de acordo com a estratégia EI foram obtidas pelo *solver* Gurobi (versão 8.1) na sua configuração padrão com tempo limite de 7200 segundos.

Tabela 33 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 15 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
3,3	209,3	301,3	576,1	1090,0	11	0,0	0,0	53,3	189,0	440,1	682,4	7	0,0
0,0	71,0	403,7	615,1	1089,8	13	0,0	0,0	26,5	197,7	386,2	610,5	6	15,4
0,0	16,1	386,8	596,4	999,3	12	0,0	0,0	32,7	215,4	385,3	633,4	6	0,0
0,0	8,2	357,3	539,9	905,4	10	0,0	0,0	27,8	200,2	406,7	634,7	6	6,8
0,0	93,4	390,7	635,2	1119,3	13	0,0	0,0	72,0	226,8	463,1	761,9	8	0,0
0,0	54,7	369,9	557,6	982,2	11	0,0	0,0	0,0	206,9	387,5	594,4	6	0,0
0,0	86,9	394,1	581,9	1062,9	12	0,0	0,0	0,0	199,2	371,5	570,7	5	3,9
0,0	0,0	296,8	492,1	788,9	9	0,0	0,0	4,6	191,7	383,3	579,6	6	0,0
0,0	0,0	316,8	538,6	855,5	11	0,0	0,0	11,0	203,3	425,2	639,5	7	0,0
0,0	33,0	397,3	628,0	1058,4	12	0,0	0,0	0,0	201,7	350,6	552,2	5	0,0
0,0	13,4	392,1	637,0	1042,6	13	0,0	0,0	8,5	233,0	404,8	646,3	7	0,0
0,0	0,0	351,8	591,4	943,2	12	0,0	0,0	0,0	210,8	360,0	570,8	5	0,0
0,0	60,1	279,1	575,0	914,2	11	2,7	0,0	67,1	215,2	491,6	773,9	9	0,0
0,0	99,4	423,0	651,4	1173,7	13	0,0	0,0	7,8	192,3	372,7	572,7	6	4,3
0,0	7,9	390,3	596,4	994,6	12	0,0	0,0	0,0	187,0	380,8	567,8	6	0,0
0,0	0,0	326,6	553,8	880,4	11	0,0	0,0	16,4	213,2	427,1	656,7	7	7,6
0,0	12,9	385,9	649,6	1048,4	14	0,0	0,0	73,8	228,4	497,7	800,0	8	0,0
0,0	13,1	386,2	583,3	982,6	12	0,0	0,0	14,1	221,8	442,7	678,5	7	0,0
0,0	44,3	406,5	573,4	1024,2	11	0,0	0,0	10,1	207,9	373,1	591,2	5	0,0
0,0	8,7	306,0	492,1	806,8	9	0,0	0,0	0,1	209,1	383,3	592,6	6	0,0
0,0	11,6	335,3	549,8	896,7	10	1,1	0,0	195,2	265,9	614,0	1075,1	11	24,3
0,0	51,7	397,7	592,8	1042,2	12	0,0	0,0	0,0	191,4	346,4	537,8	5	6,5
0,0	24,4	391,8	650,2	1066,4	14	0,0	0,0	3,6	228,5	422,3	654,4	6	9,5
0,0	9,8	420,9	602,2	1032,8	12	0,0	0,0	0,0	235,2	360,0	595,2	4	4,8
0,0	160,5	295,5	575,0	1031,1	11	10,2	0,0	25,9	193,3	460,9	680,2	8	0,0
0,0	111,1	425,6	617,8	1154,4	13	0,0	0,0	30,5	181,9	385,3	597,7	6	9,3
0,0	13,7	395,8	596,4	1005,9	12	0,0	0,0	3,9	200,2	383,8	588,0	6	9,2
0,0	24,3	347,7	506,9	878,9	10	0,0	0,0	14,8	205,4	394,7	615,0	6	20,4
0,0	252,7	432,0	675,2	1360,0	14	11,7	0,0	16,7	213,3	450,5	680,5	7	0,0
0,0	169,0	416,6	588,1	1173,7	12	0,0	0,0	0,0	206,8	388,3	595,1	6	0,0
0,0	26,5	411,4	581,9	1019,8	12	0,0	0,0	3,5	186,3	377,3	567,1	5	1,8
0,0	4,4	313,0	506,0	823,5	9	0,0	0,0	7,7	192,0	361,6	561,2	5	0,0
0,0	35,8	332,9	546,4	915,1	9	0,0	0,0	20,6	225,2	438,8	684,6	8	6,7
0,0	1,3	388,3	592,8	982,4	12	0,0	0,0	3,0	204,6	350,6	558,2	5	0,0
0,0	15,4	409,4	611,0	1035,8	12	1,4	0,0	14,2	228,0	395,1	637,3	6	3,6
0,0	3,4	403,1	591,4	998,0	12	0,0	0,0	0,0	219,2	360,0	579,2	5	0,0
0,1	48,6	371,6	584,7	1005,0	12	0,8	0,0	21,3	209,1	406,2	636,6	6	3,7

Fonte: Elaborada pelo autor.

Nas Tabelas 33, 34 e 35, temos as informações sobre soluções obtidas com a estratégia EI para os conjuntos de instâncias com 15, 25 e 40 clientes, respectivamente. Por conta do espaço, não são colocados os nomes das instâncias nas tabelas, mas a disposição vertical das informações de acordo com cada instância é análoga às tabelas apresentadas nas seções anteriores, lembrando que, de cima para baixo, estão as informações referentes às soluções das instâncias R101-R112 do Grupo A, depois das instâncias R101-R112 do Grupo B e, por último, das instâncias R101-R112 do Grupo C. Além disso, temos também a média dos valores de cada coluna apresentada na última linha em negrito.

Tabela 34 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 25 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
10,2	78,2	404,0	1039,8	1532,1	18	0,0	3,7	60,5	229,0	955,2	955,2	10	15,5
3,5	6,1	442,7	1046,3	1498,6	20	0,4	6,4	5,6	214,6	843,6	843,6	9	14,0
0,0	3,0	433,7	1049,8	1486,6	19	2,0	0,0	10,6	205,3	797,1	797,1	8	21,7
0,0	4,3	422,7	963,7	1390,7	18	2,3	0,0	0,0	209,7	828,2	828,2	9	21,4
0,0	24,5	445,0	1076,1	1545,6	20	3,4	0,0	6,7	243,0	900,4	900,4	10	9,1
0,0	3,4	407,2	1010,6	1421,1	18	4,9	1,0	22,1	241,6	911,2	911,2	10	12,2
0,0	0,0	441,3	1029,5	1470,9	19	0,5	0,0	0,0	214,2	839,3	839,3	9	18,2
0,0	0,0	458,9	1016,3	1475,2	19	2,3	0,0	12,7	240,3	878,1	878,1	9	16,6
0,0	3,2	369,9	904,4	1277,4	16	4,0	2,2	4,2	197,6	767,2	767,2	8	15,6
0,0	0,0	429,9	1008,5	1438,4	19	0,9	0,0	0,0	230,6	847,6	847,6	9	16,2
0,0	12,9	393,3	986,8	1393,0	19	2,4	0,0	0,0	266,5	921,5	921,5	10	9,4
0,0	0,0	429,2	1101,3	1530,5	20	2,9	0,0	1,9	276,7	911,4	911,4	9	11,0
0,0	79,5	387,4	1031,6	1498,4	17	13,7	0,0	66,1	242,4	987,3	987,3	10	20,4
3,5	0,0	466,9	1054,7	1525,1	19	3,9	0,0	26,7	218,7	854,5	854,5	9	16,5
0,0	0,0	431,5	1049,8	1481,4	19	3,1	0,0	1,9	214,2	807,9	807,9	9	24,0
0,0	11,2	428,3	985,5	1425,0	18	9,4	1,5	0,0	208,2	833,4	833,4	9	20,6
0,0	3,7	460,5	1076,1	1540,3	20	5,1	0,1	11,5	244,1	903,0	903,0	10	11,2
0,0	0,0	412,2	1018,0	1430,2	19	5,5	0,0	1,0	237,7	899,6	899,6	10	14,7
0,0	0,0	448,2	1029,5	1477,7	19	4,0	0,0	0,0	225,4	828,9	828,9	8	17,0
0,0	0,2	449,2	1016,3	1465,7	19	4,4	0,0	0,7	228,6	849,3	849,3	9	18,3
0,0	30,5	394,4	904,4	1329,2	16	12,3	0,0	4,2	208,9	788,1	788,1	9	18,1
0,0	0,0	439,1	1008,5	1447,6	18	3,9	0,2	0,0	228,9	867,8	867,8	9	25,9
0,0	177,8	487,5	1084,4	1749,7	21	5,6	0,0	0,2	268,6	921,2	921,2	10	15,3
0,0	0,0	439,4	1103,2	1542,6	20	4,3	0,0	0,0	241,2	870,5	870,5	9	16,4
0,0	230,4	417,4	1029,0	1676,9	18	23,9	0,0	56,1	244,9	994,5	994,5	11	22,2
3,5	23,6	495,8	1046,3	1569,2	20	9,4	0,0	47,4	219,7	882,7	882,7	9	19,2
0,0	88,5	500,0	1074,2	1662,7	20	6,3	7,0	4,3	206,8	807,8	807,8	8	23,2
0,0	14,2	457,9	964,3	1436,5	17	2,5	5,0	11,9	214,4	830,9	830,9	8	19,7
0,0	56,6	474,2	1106,1	1636,9	21	13,7	0,0	25,2	252,3	931,9	931,9	10	0,0
0,0	0,1	446,9	1018,0	1465,0	19	5,5	0,0	7,5	246,8	887,7	887,7	10	17,3
0,0	0,0	462,0	1029,5	1491,5	19	5,7	0,0	0,0	214,9	809,0	809,0	8	21,1
0,0	5,6	458,2	1016,4	1480,2	19	3,1	1,0	0,0	229,1	859,9	859,9	9	23,0
0,0	2,9	380,5	914,8	1298,2	15	9,5	1,2	3,0	209,4	786,9	786,9	8	21,9
0,0	28,0	474,8	1008,5	1511,3	18	13,0	0,0	0,2	225,3	863,2	863,2	9	22,8
0,0	27,8	425,5	986,1	1439,4	18	8,2	0,0	2,1	264,8	931,8	931,8	10	16,8
0,0	0,0	463,5	1101,3	1564,8	20	11,6	0,0	0,0	236,1	865,5	865,5	9	21,0
0,6	25,5	438,3	1024,7	1489,0	19	5,9	0,8	10,9	230,6	868,4	868,4	9	17,4

Fonte: Elaborada pelo autor.

Nas três tabelas, reportamos os tempos que compõem a função objetivo de cada solução separadamente. Assim, temos o total de tempo de antecipação (AN), o total de tempo de atraso (AT), o *makespan* (MS) e o total de tempo de viagem (TV). Essas quatro parciais somadas

forneem o valor da função objetivo (VF) de cada solução. Além disso, reportamos também o número de veículos utilizados (NV) e o *gap* de otimalidade em percentagem de cada solução (GAP). Em especial, na Tabela 35 no caso do Cenário 2, observe que existem alguns traços ao invés de valores, pois, para estas instâncias, não foram obtidas soluções no tempo limite estabelecido.

Tabela 35 – Informações sobre as soluções obtidas com estratégia integrada para as instâncias com 40 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
11,1	209,4	446,8	1589,2	2256,5	28	17,3	13,3	196,8	270,6	1106,9	1587,6	18	30,9
0,0	134,2	495,6	1706,0	2335,8	32	10,4	-	-	-	-	-	-	-
0,0	6,0	461,0	1522,1	1989,1	27	6,1	37,1	75,7	272,6	1124,1	1509,4	17	37,3
0,0	2,1	486,2	1592,8	2081,2	30	5,2	0,2	41,6	271,8	1075,4	1389,0	16	31,3
0,0	61,3	483,4	1744,4	2289,0	32	11,9	18,5	153,3	257,3	1162,5	1591,6	19	38,9
0,0	54,7	497,4	1741,0	2293,1	32	10,7	0,6	6,0	240,5	1027,3	1274,4	15	30,9
0,0	79,9	424,9	1420,0	1924,7	26	1,3	7,5	51,3	243,9	979,9	1282,6	14	35,9
0,0	0,0	418,4	1459,4	1877,8	25	3,4	-	-	-	-	-	-	-
0,0	2,7	484,3	1642,1	2129,0	31	5,2	0,0	69,6	264,8	1076,0	1410,4	15	36,9
0,0	3,0	424,5	1419,8	1847,2	26	5,0	0,1	19,3	262,6	1019,3	1301,3	15	35,1
0,0	0,0	437,4	1596,7	2034,1	28	2,7	0,0	88,7	229,4	1068,2	1386,4	14	45,6
0,0	0,9	472,9	1605,7	2079,6	29	2,4	0,0	0,0	251,3	1015,3	1266,6	14	32,8
0,0	298,7	460,7	1561,2	2320,6	27	17,0	0,0	351,4	277,8	1127,9	1757,1	18	40,0
0,0	166,0	500,4	1752,5	2418,9	32	14,6	0,4	169,0	245,4	1082,6	1497,3	18	39,7
0,0	139,5	481,2	1564,8	2185,5	29	12,1	14,6	299,0	298,8	1164,8	1777,2	1	45,3
0,0	14,5	503,8	1603,5	2121,8	30	6,0	0,0	3,0	262,8	1095,8	1361,7	17	34,0
0,0	235,3	474,2	1794,2	2503,8	33	18,5	61,0	164,9	251,2	1151,4	1628,4	16	41,8
0,0	137,8	522,3	1719,7	2379,8	32	13,4	0,0	102,9	256,7	1053,9	1413,5	17	35,8
0,0	76,8	425,4	1461,3	1963,5	27	10,2	0,0	174,6	239,7	1074,6	1489,0	15	44,3
0,0	0,0	443,2	1460,8	1904,0	25	4,4	0,0	39,5	220,6	1002,4	1262,5	13	39,4
0,0	113,9	505,5	1694,5	2314,0	32	12,8	2,6	49,8	265,7	1147,3	1465,4	19	38,3
0,0	13,4	429,6	1419,8	1862,7	26	7,4	7,7	5,6	250,6	1090,5	1354,4	16	37,8
0,0	49,6	476,6	1628,0	2154,2	29	14,0	0,0	1,8	222,3	1010,8	1234,8	14	37,4
0,0	1,1	497,4	1601,6	2100,1	29	7,6	0,0	1,1	248,1	1098,7	1347,9	17	40,1
6,5	706,1	476,3	1648,0	2836,9	29	35,9	-	-	-	-	-	-	-
0,0	260,1	504,7	1738,4	2503,3	33	20,3	0,0	523,9	263,0	1152,8	1939,8	16	53,8
7,0	165,6	501,5	1594,1	2268,1	29	13,4	17,0	150,4	269,2	1135,6	1572,3	17	39,2
0,0	71,5	510,3	1683,0	2264,9	31	17,9	0,0	9,0	269,6	1129,8	1408,4	16	35,7
0,0	772,4	511,6	1780,6	3064,6	32	34,7	-	-	-	-	-	-	-
0,0	215,6	523,2	1745,9	2484,7	33	18,9	16,1	213,9	262,1	1071,8	1563,9	16	45,0
0,0	224,1	431,0	1514,5	2169,6	27	22,8	-	-	-	-	-	-	-
0,0	19,3	460,9	1459,4	1939,6	25	5,8	-	-	-	-	-	-	-
0,0	694,4	526,6	1728,8	2949,9	34	29,2	-	-	-	-	-	-	-
0,0	55,8	453,3	1498,6	2007,6	27	12,9	-	-	-	-	-	-	-
0,0	61,0	487,2	1639,4	2187,6	29	13,8	19,6	68,2	216,8	979,3	1283,9	12	40,1
0,0	0,0	488,8	1601,6	2090,4	29	10,8	-	-	-	-	-	-	-
0,7	140,2	475,8	1609,3	2225,9	29	12,7	8,0	112,2	255,0	1082,4	1457,6	15	38,6

Fonte: Elaborada pelo autor.

A.3.2 Estratégia roteamento/cross-dock

Nas Tabelas 36, 37 e 38, são reportadas as informações sobre soluções obtidas com a estratégia RCD para os conjuntos de instâncias com 15, 25 e 40 clientes, respectivamente. Nas tabelas, temos basicamente as mesmas informações apresentadas nas tabelas da seção anterior para o caso da estratégia EI.

Tabela 36 – Informações sobre as soluções obtidas com a estratégia RCD - 15 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						R	CD							R	CD
0,0	584,6	316,1	511,1	1411,8	8	0,0	0,0	0,0	249,4	208,2	378,8	836,5	5	0,0	0,0
0,0	69,4	405,8	614,9	1090,1	12	0,0	0,0	0,0	128,8	199,9	352,2	680,9	5	0,0	0,0
0,0	31,8	380,5	596,4	1008,6	12	0,0	0,0	0,0	121,0	220,7	346,3	687,9	5	0,0	0,0
0,0	44,6	359,8	506,9	911,3	10	0,0	0,0	0,0	177,3	216,2	339,7	733,1	5	0,0	0,0
0,0	80,0	422,8	622,6	1125,5	12	0,0	0,0	0,0	240,8	240,1	384,3	865,2	6	0,0	0,0
0,0	90,8	365,6	548,0	1004,4	11	0,0	0,0	0,0	42,4	206,2	372,7	621,2	6	0,0	0,0
0,0	116,7	401,2	573,4	1091,3	11	0,0	0,0	0,0	110,7	198,2	369,2	678,2	5	0,0	0,0
0,0	0,0	296,8	492,1	788,9	9	0,0	0,0	0,0	72,9	199,7	358,1	630,7	5	0,0	0,0
0,0	41,8	330,3	535,1	907,3	9	0,0	0,0	0,0	120,9	210,1	361,6	692,6	6	0,0	0,0
-	-	-	-	-	-	-	-	0,0	14,7	201,7	346,4	562,7	5	0,0	0,0
0,0	115,6	396,7	594,2	1106,5	12	0,0	0,0	0,0	51,0	242,1	385,9	678,9	6	0,0	0,0
0,0	0,0	351,8	591,4	943,2	12	0,0	0,0	0,0	0,0	210,8	360,0	570,8	5	0,0	0,0
0,0	216,2	288,2	511,1	1015,6	8	0,0	0,0	-	-	-	-	-	-	-	-
0,0	173,2	434,6	614,9	1222,7	12	0,0	0,0	0,0	95,7	201,5	352,2	649,4	5	0,0	0,0
0,0	11,0	390,4	596,4	997,8	12	0,0	0,0	0,0	47,3	200,6	346,3	594,2	5	0,0	0,0
0,0	10,9	362,8	506,9	880,7	10	0,0	0,0	0,0	224,7	221,6	339,7	785,9	5	0,0	0,0
0,0	57,2	386,2	622,6	1066,1	12	0,0	0,0	0,0	328,3	240,6	384,3	953,2	6	0,0	0,0
0,0	119,2	386,2	548,0	1053,4	11	0,0	0,0	0,0	94,4	239,8	372,7	706,8	6	0,0	0,0
0,0	44,4	406,5	573,4	1024,3	11	0,0	0,0	0,0	203,0	211,7	369,2	783,9	5	0,0	0,0
0,0	8,7	306,0	492,1	806,8	9	0,0	0,0	0,0	103,6	217,1	358,1	678,8	5	0,0	0,0
0,0	105,3	340,8	535,1	981,2	9	0,0	0,0	-	-	-	-	-	-	-	-
0,0	122,3	406,8	584,0	1113,1	11	0,0	0,0	0,0	38,2	191,4	346,4	576,0	5	0,0	0,0
-	-	-	-	-	-	-	-	0,0	46,9	244,7	385,9	677,4	6	0,0	0,0
0,0	49,9	427,6	591,4	1068,9	12	0,0	0,0	0,0	0,0	235,2	360,0	595,2	5	0,0	0,0
0,0	406,1	304,2	511,1	1221,4	8	0,0	0,0	0,0	268,9	206,0	378,8	853,7	5	0,0	0,0
0,0	129,0	437,0	614,9	1181,0	12	0,0	0,0	0,0	61,7	192,0	352,2	605,8	5	0,0	0,0
0,0	32,9	395,8	596,4	1025,1	12	0,0	0,0	0,0	93,6	207,3	346,3	647,1	5	0,0	0,0
0,0	24,3	347,7	506,9	878,9	10	0,0	0,0	0,0	106,0	205,8	339,7	651,5	5	0,0	0,0
0,0	366,9	426,4	622,6	1415,9	12	0,0	0,0	0,0	152,1	218,8	384,3	755,2	6	0,0	0,0
0,0	296,8	404,4	548,0	1249,2	11	0,0	0,0	0,0	23,9	206,8	372,7	603,3	6	0,0	0,0
0,0	63,0	418,4	573,4	1054,8	11	0,0	0,0	0,0	100,6	201,6	369,2	671,5	5	0,0	0,0
0,0	20,0	321,6	492,1	833,6	9	0,0	0,0	0,0	40,6	193,2	358,1	591,9	5	0,0	0,0
0,0	197,6	339,0	535,1	1071,7	9	0,0	0,0	0,0	183,3	225,6	361,6	770,5	6	0,0	0,0
0,0	65,8	388,1	584,0	1037,9	11	0,0	0,0	0,0	65,7	204,6	346,4	616,7	5	0,0	0,0
0,0	108,9	410,6	594,2	1113,6	12	0,0	0,0	0,0	34,4	235,2	385,9	655,4	6	0,0	0,0
0,0	16,5	408,2	591,4	1016,1	12	0,0	0,0	0,0	0,0	219,2	360,0	579,2	5	0,0	0,0
0,0	112,4	375,4	562,7	1050,5	11	0,0	0,0	0,0	107,1	214,0	362,5	683,6	5	0,0	0,0

Fonte: Elaborada pelo autor.

Vale a pena lembrar que, na estratégia RCD, são resolvidos os problemas de roteamento e de programação do *cross-dock* separadamente e, por isso, são reportados dois valores na coluna em que estão os *gaps* de otimalidade. Primeiro, o problema de roteamento é resolvido para obter as decisões de consolidação e de ordem das visitas. Na sequência, com essas decisões definidas,

o problema de programação do *cross-dock* é também resolvido. Dessa forma, o primeiro valor é o *gap* da solução obtida ao resolver o problema de roteamento (R) e o segundo valor é o *gap* da solução obtida ao resolver o problema de programação do *cross-dock* (CD). Vale ressaltar que tempo de solução para cada um desses problemas é limitado a uma hora.

Tabela 37 – Informações sobre as soluções obtidas com a estratégia RCD - 25 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						R	CD							R	CD
0,0	381,9	417,1	985,5	1784,5	16	0,0	0,0	0,0	116,5	246,4	662,5	1025,4	9	0,0	0,0
3,5	7,0	441,7	1046,3	1498,6	20	0,0	0,0	0,0	29,9	207,3	602,3	839,5	9	0,0	0,0
0,0	1,6	432,4	1049,8	1483,8	19	0,0	1,7	7,0	17,3	206,0	549,3	779,6	8	9,3	0,0
0,0	8,3	433,7	959,0	1401,0	17	0,0	1,6	0,0	52,1	212,3	581,1	845,4	8	6,2	0,0
0,0	16,7	451,0	1076,1	1543,8	20	0,0	3,1	0,0	52,2	259,5	642,7	954,5	9	0,0	0,0
0,0	3,4	412,2	1010,6	1426,1	18	0,0	0,0	3,0	54,1	241,4	619,4	917,9	9	5,6	0,0
0,0	0,0	441,3	1029,3	1470,6	19	0,0	0,0	0,0	6,7	215,3	584,1	806,1	8	6,5	0,0
0,0	0,0	458,9	1016,3	1475,2	19	0,0	1,9	0,0	51,6	246,6	610,8	908,9	9	13,6	0,0
0,0	4,8	363,5	904,4	1272,7	16	0,0	0,0	0,0	0,0	205,4	563,8	769,2	8	9,5	0,0
0,0	0,0	435,6	1008,5	1444,2	18	0,0	1,2	0,0	5,8	236,6	604,6	847,0	8	11,3	0,0
0,0	8,2	410,9	986,1	1405,2	18	0,0	0,9	0,0	3,6	258,0	652,3	913,9	10	0,0	0,0
0,0	0,0	427,7	1101,3	1529,0	20	0,0	2,3	0,0	58,7	276,7	627,7	963,1	9	8,5	0,0
10,2	158,0	381,2	985,5	1534,9	16	0,0	4,6	0,0	130,6	251,3	662,5	1044,4	9	0,0	0,0
3,5	4,5	478,0	1046,3	1532,3	20	0,0	3,5	0,0	35,9	218,0	602,3	856,2	9	1,8	0,0
0,0	0,0	433,1	1049,8	1482,9	19	0,0	2,1	0,0	36,8	219,4	566,3	822,5	8	10,7	2,0
0,0	35,0	432,7	959,0	1426,7	17	0,0	2,8	0,0	41,7	217,6	581,1	840,4	8	6,3	0,0
0,0	1,1	450,9	1076,1	1528,1	20	0,0	3,6	0,0	24,6	251,8	642,7	919,1	9	0,0	0,0
0,0	33,6	414,7	1010,6	1458,9	18	0,0	1,4	1,1	26,3	255,6	619,4	902,4	9	5,9	0,0
0,0	17,4	450,0	1029,3	1496,7	19	0,0	3,2	0,0	0,0	222,6	583,4	806,0	8	5,4	0,0
0,0	0,0	448,2	1016,3	1464,5	19	0,0	3,2	0,0	0,0	230,4	610,8	841,2	9	13,2	0,0
0,0	0,6	377,0	904,4	1282,0	16	0,0	6,6	0,0	0,0	206,6	563,8	770,4	8	9,5	0,0
0,0	0,0	439,1	1008,5	1447,6	18	0,0	2,6	0,0	34,8	230,7	598,4	863,9	8	10,4	0,0
-	-	-	-	-	-	-	-	0,0	2,5	268,6	652,3	923,3	10	0,0	0,0
0,0	0,0	443,0	1101,3	1544,3	20	0,0	2,8	0,0	0,0	240,3	628,9	869,2	9	9,8	0,0
0,0	362,0	413,5	985,5	1761,0	16	0,0	28,2	4,1	134,2	242,9	662,5	1043,7	9	0,0	2,4
3,5	32,6	485,8	1046,3	1568,2	20	0,0	9,7	0,0	49,1	220,1	602,3	871,5	9	1,8	8,3
0,0	177,3	499,6	1049,8	1726,7	19	0,0	5,5	7,0	13,0	207,7	549,3	777,0	8	9,0	0,0
0,0	56,8	457,9	959,0	1473,7	17	0,0	2,3	0,0	15,4	213,2	581,1	809,7	8	6,2	1,0
0,0	84,2	475,5	1076,1	1635,8	20	0,0	16,8	0,0	85,2	257,4	642,7	985,4	9	0,0	7,8
0,0	47,9	451,7	1010,6	1510,2	18	0,0	6,6	3,0	15,3	252,2	619,4	890,0	9	5,9	0,0
0,0	1,4	459,3	1029,3	1490,0	19	0,0	3,9	0,0	0,0	216,5	583,4	799,9	8	6,2	0,4
0,0	5,4	458,0	1016,3	1479,7	19	0,0	1,8	0,0	12,3	228,7	604,1	845,0	9	11,8	0,0
0,0	7,8	397,4	904,4	1309,6	16	0,0	6,3	0,0	22,6	208,7	565,4	796,7	8	9,8	7,9
0,0	2,1	475,7	1008,5	1486,3	18	0,0	14,4	0,0	17,0	227,4	605,3	849,7	8	10,7	2,6
0,0	14,0	439,8	986,1	1440,0	18	0,0	7,2	0,0	2,4	270,9	652,3	925,5	10	0,0	0,7
0,0	0,0	463,3	1101,3	1564,6	20	0,0	9,7	0,0	0,0	244,7	627,7	872,4	9	8,0	1,5
0,6	42,1	438,6	1015,2	1496,6	18	0,0	4,6	0,7	31,9	233,7	608,6	874,9	9	5,9	1,0

Fonte: Elaborada pelo autor.

Resolvendo o problema de roteamento primeiro, sem levar em consideração as decisões para a programação do *cross-dock*, verificamos que é possível obter rotas que são infactíveis para o sistema, ou seja, não é possível consolidar os pedidos dos clientes que estão numa mesma rota, distribuí-los e retornar com o veículo ao *cross-dock* antes do tempo limite estabelecido (T^{max}). Em outras palavras, com determinadas rotas, não é possível obter uma planejamento viável para as operações no *cross-dock*, isto é, o problema de programação do *cross-dock* é infactível. Assim,

quando isso acontece, consideramos que a estratégia RCD não fornece uma solução viável para o sistema de distribuição. Por isso, nas Tabelas 36, 37 e 38, existem alguns campos com traços para as instâncias cujas soluções obtidas ao resolver do problema de roteamento apresentaram consolidação de cargas que tornam o problema de programação do *cross-dock* infactível.

Tabela 38 – Informações sobre as soluções obtidas com a estratégia RCD - 40 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						R	CD							R	CD
0,0	256,9	440,6	1537,0	2234,4	26	0,0	13,2	4,2	207,7	265,2	1021,8	1498,8	16	13,4	10,5
0,0	107,2	491,8	1691,6	2290,6	31	0,0	8,5	0,0	180,1	248,9	949,7	1378,7	15	19,5	10,6
0,0	16,3	446,6	1519,9	1982,8	27	0,0	5,8	0,0	113,0	264,2	978,1	1355,2	15	21,4	11,1
0,0	2,7	469,5	1592,8	2065,0	30	0,0	4,4	0,0	66,1	279,9	977,2	1323,3	14	24,4	4,9
0,0	114,6	478,2	1693,2	2285,9	31	0,0	11,6	0,0	153,0	251,2	981,5	1385,7	15	19,9	9,9
0,0	85,3	494,4	1700,9	2280,6	31	0,0	10,2	0,0	148,1	238,3	915,2	1301,6	14	20,7	8,9
0,0	76,9	424,9	1420,0	1921,8	26	0,0	1,0	0,0	109,3	243,5	880,5	1233,3	13	24,8	12,9
0,0	0,0	420,0	1459,4	1879,4	25	0,0	2,6	-	-	-	-	-	-	-	-
0,0	1,5	489,7	1629,1	2120,3	30	0,0	4,8	0,0	36,2	251,0	985,6	1272,9	15	26,2	7,4
0,0	0,0	419,7	1419,8	1839,4	26	0,0	4,6	0,0	0,0	244,3	956,2	1200,5	14	28,6	3,5
0,0	0,0	443,7	1596,7	2040,4	28	0,0	3,0	2,0	0,0	224,8	847,5	1074,3	13	25,9	3,9
0,0	0,0	456,7	1601,6	2058,3	29	0,0	1,4	0,0	0,0	240,7	885,2	1125,9	14	24,0	0,0
0,0	369,6	449,9	1537,0	2356,5	26	0,0	18,4	0,0	209,9	266,2	1019,4	1495,5	16	11,8	19,3
0,0	164,2	492,6	1691,6	2348,4	31	0,0	11,8	3,8	129,9	242,8	949,6	1326,2	14	19,1	15,1
0,0	76,8	485,5	1519,9	2082,3	27	0,0	7,1	0,0	217,3	274,1	964,9	1456,3	15	19,7	13,6
0,0	0,0	499,8	1592,8	2092,7	30	0,0	4,5	0,0	3,6	257,7	1000,3	1261,6	15	25,4	2,1
0,0	293,6	488,8	1693,2	2475,6	31	0,0	17,6	0,0	128,5	248,8	974,1	1351,4	14	17,5	14,5
0,0	88,7	507,7	1700,9	2297,4	31	0,0	10,2	0,0	139,8	258,2	920,5	1318,5	14	21,2	2,9
0,0	82,4	423,9	1420,0	1926,2	26	0,0	6,8	0,0	97,3	236,2	914,8	1248,3	14	28,4	8,3
0,0	0,0	430,8	1459,4	1890,2	25	0,0	3,7	0,0	7,1	221,5	845,3	1073,9	12	28,5	3,6
0,0	50,7	492,0	1629,1	2171,8	30	0,0	6,2	0,0	27,5	257,3	985,6	1270,5	15	26,2	5,8
0,0	11,4	440,8	1419,8	1872,0	26	0,0	7,8	0,0	0,0	246,2	925,9	1172,1	14	26,2	4,5
0,0	29,3	468,6	1596,7	2094,6	28	0,0	11,5	2,0	23,1	217,6	847,5	1090,3	13	25,9	6,2
0,0	0,0	473,0	1601,6	2074,6	29	0,0	6,5	0,0	0,0	237,8	885,2	1123,0	14	24,0	3,0
0,0	729,9	473,6	1537,0	2740,5	26	0,0	33,6	2,0	335,5	272,6	1033,5	1643,6	16	12,8	27,0
0,0	335,3	498,5	1691,6	2525,4	31	0,0	21,0	3,8	240,6	247,2	948,4	1440,0	14	20,2	22,1
0,0	199,8	494,2	1519,9	2214,0	27	0,0	6,7	0,0	229,6	274,4	970,3	1474,3	15	20,6	12,4
0,0	28,4	493,0	1592,8	2114,2	30	0,0	12,0	0,0	30,2	266,3	983,2	1279,7	15	24,1	7,0
0,0	491,0	500,4	1693,2	2684,6	31	0,0	25,4	0,0	147,9	256,1	1010,7	1414,7	15	20,8	13,9
0,0	505,7	519,9	1700,9	2726,6	31	0,0	26,1	0,0	118,2	252,2	909,9	1280,3	14	20,2	16,5
0,0	64,5	426,7	1420,0	1911,2	26	0,0	11,6	0,0	98,0	246,0	951,4	1295,3	13	30,1	14,3
0,0	8,0	443,8	1459,4	1911,2	25	0,0	4,1	0,0	4,8	212,3	841,2	1058,3	12	28,2	4,0
0,0	809,5	524,7	1629,1	2963,3	30	0,0	24,1	0,0	8,7	264,9	962,0	1235,5	14	24,2	9,5
0,0	38,5	434,6	1419,8	1892,8	26	0,0	7,6	0,0	39,1	257,6	926,8	1223,5	14	26,3	12,3
0,0	11,9	474,3	1596,7	2083,0	28	0,0	9,5	0,0	38,7	214,7	838,9	1092,3	12	24,8	7,4
0,0	0,0	478,0	1601,6	2079,7	29	0,0	10,3	0,0	0,0	247,2	893,3	1140,5	14	26,1	8,1
0,0	140,3	469,2	1571,8	2181,3	28	0,0	10,4	0,5	94,0	249,4	939,5	1283,3	14	22,9	9,6

Fonte: Elaborada pelo autor.

A.3.3 Estratégia cross-dock/roteamento

Nas Tabelas 39, 40 e 41, reportamos as informações sobre as soluções obtidas com as estratégia CDR para os conjuntos de instâncias com 15, 25 e 40 clientes, respectivamente.

Tabela 39 – Informações sobre as soluções obtidas com a estratégia CDR - 15 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						CD	R							CD	R
0,0	209,3	301,3	636,8	1147,5	13	0,0	0,0	0,0	72,7	187,4	542,5	802,5	8	0,0	0,0
0,0	53,4	403,7	678,9	1136,0	14	0,0	0,0	0,0	28,1	191,1	517,6	736,8	8	0,0	0,0
0,0	3,9	384,8	663,6	1052,4	14	0,0	0,0	0,0	4,4	201,9	522,0	728,3	9	0,0	0,0
0,0	11,8	372,4	583,3	967,5	12	0,0	0,0	0,0	5,8	196,4	481,6	683,8	9	0,0	0,0
0,0	90,4	384,0	656,7	1131,1	14	0,0	0,0	0,0	40,0	218,8	562,6	821,5	11	0,0	0,0
0,0	38,0	364,0	606,4	1008,4	13	0,0	0,0	0,0	0,0	202,7	496,3	699,0	9	0,0	0,0
0,0	78,2	390,7	650,3	1119,2	13	0,0	0,0	0,0	6,6	186,7	443,7	637,0	6	0,0	0,0
0,0	0,0	289,6	591,8	881,5	11	0,0	0,0	0,0	8,5	191,2	447,9	647,6	7	0,0	0,0
0,0	0,0	317,2	623,7	940,9	12	0,0	0,0	0,0	25,7	215,6	487,4	728,8	7	0,0	0,0
0,0	40,3	397,3	656,7	1094,3	14	0,0	0,0	0,0	0,0	192,8	472,4	665,2	6	0,0	0,0
0,0	42,5	385,8	628,9	1057,2	13	0,0	0,0	0,0	0,7	230,1	526,7	757,5	8	0,0	0,0
0,0	0,0	355,3	627,8	983,1	13	0,0	0,0	0,0	0,0	205,0	456,3	661,3	7	0,0	0,0
0,0	51,5	290,7	603,5	945,7	11	0,0	0,0	0,0	49,4	217,4	559,3	826,1	10	0,0	0,0
0,0	138,2	423,0	615,1	1176,3	13	0,0	0,0	0,0	17,5	189,6	517,5	724,6	8	0,0	0,0
0,0	16,4	386,3	624,2	1026,9	13	0,0	0,0	0,0	0,3	196,8	466,8	663,9	6	1,5	0,0
0,0	7,4	338,1	588,1	933,6	11	0,0	0,0	0,0	0,0	210,2	528,2	738,4	9	0,0	0,0
0,0	27,1	385,9	656,7	1069,7	14	0,0	0,0	0,0	87,6	235,1	532,2	854,9	10	0,0	0,0
0,0	13,1	374,1	657,8	1045,1	14	0,0	0,0	0,0	2,2	223,4	515,0	740,6	9	0,0	0,0
0,0	48,7	384,4	632,2	1065,2	13	0,0	0,0	0,0	16,5	206,8	425,5	648,9	6	0,0	0,0
0,0	0,0	297,3	575,6	872,9	12	0,0	0,0	0,0	4,5	211,8	486,0	702,3	8	0,4	0,0
0,0	0,7	317,2	615,2	933,1	12	0,0	0,0	0,0	182,8	265,9	630,2	1079,0	12	0,0	0,0
0,0	56,0	379,7	657,5	1093,2	14	0,0	0,0	0,0	0,0	189,3	463,6	652,9	6	0,0	0,0
0,0	24,4	393,8	650,2	1068,4	14	0,0	0,0	0,0	0,0	236,2	493,6	729,8	8	0,0	0,0
0,0	0,0	410,9	646,1	1057,0	14	0,0	0,0	0,0	0,0	213,0	477,3	690,3	8	0,0	0,0
0,0	172,2	290,9	607,4	1070,5	12	0,0	0,0	0,0	23,2	197,3	520,4	740,9	9	0,0	0,0
0,0	93,4	423,0	679,0	1195,5	15	0,0	0,0	0,0	9,7	182,4	491,3	683,4	8	0,9	0,0
0,0	2,0	400,9	634,2	1037,1	13	0,0	0,0	0,0	14,2	199,9	453,5	667,7	7	0,0	0,0
0,0	17,0	340,6	577,1	934,7	12	0,0	0,0	0,0	60,2	213,4	398,3	671,9	6	0,4	0,0
0,0	262,3	422,0	679,0	1363,4	15	0,0	0,0	0,0	42,4	216,6	504,3	763,3	9	0,1	0,0
0,0	170,8	405,0	629,4	1205,2	13	0,0	0,0	0,0	1,0	205,2	477,9	684,1	8	1,6	0,0
0,0	32,3	407,6	648,7	1088,6	13	0,0	0,0	0,0	1,5	190,3	438,9	630,8	7	0,9	0,0
0,0	20,0	324,4	557,4	901,8	9	1,4	0,0	0,0	5,4	192,4	477,8	675,6	6	0,5	0,0
0,0	20,9	322,1	621,3	964,3	13	0,0	0,0	4,6	25,7	221,3	491,1	742,7	9	0,0	0,0
0,0	0,0	384,2	625,4	1009,6	13	0,0	0,0	0,0	6,8	193,1	456,6	656,4	7	1,2	0,0
0,0	0,0	419,4	631,4	1050,8	13	1,2	0,0	0,0	6,0	224,5	527,7	758,2	7	0,0	0,0
0,0	12,3	404,2	609,8	1026,3	12	0,0	0,0	0,0	0,0	205,3	471,8	677,1	7	0,4	0,0
0,0	48,7	368,7	628,5	1045,9	13	0,1	0,0	0,1	20,8	207,1	493,4	721,5	8	0,2	0,0

Fonte: Elaborada pelo autor.

Nas tabelas, temos as mesmas informações das tabelas apresentadas no caso da estratégia RCD. Lembrando que a ordem de resolução dos dois problemas é invertida. Primeiro, o problema de programação do *cross-dock* (CD) é resolvido admitindo apenas entregas diretas. Com a solução obtida, definimos as decisões para as operações de descarregamento e as docas em que cada cliente é servido (i.e, a doca em que o pedido do cliente é carregado). Na sequência, de acordo com essas decisões obtidas com a solução do primeiro problema, o problema roteamento

(R) é resolvido permitindo a consolidação de cargas com pedidos de clientes apenas servidos na mesma doca e estabelecendo a ordem do carregamento das cargas de saída consolidadas. Assim, dois valores também são apresentados na coluna dos *gaps* de otimalidade, o primeiro valor é o *gap* da solução obtida com a resolução do primeiro problema (CD) e o segundo valor é o *gap* da solução obtida com a resolução do segundo problema (R). O tempo de solução para cada um dos problemas é também limitado a uma hora.

Tabela 40 – Informações sobre as soluções obtidas com a estratégia CDR - 25 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						CD	R							CD	R
0,0	81,0	400,9	1194,3	1676,2	22	0,0	0,0	0,7	41,1	251,1	857,1	1149,9	14	0,2	0,0
0,0	4,9	456,0	1140,3	1601,2	21	0,2	0,0	7,7	25,8	210,2	819,2	1062,9	12	0,3	0,0
0,0	4,7	436,7	1157,0	1598,4	22	1,4	0,0	0,0	1,9	212,2	762,4	976,6	10	0,0	0,0
0,0	5,2	429,2	1087,3	1521,7	19	0,4	0,0	11,0	13,9	213,7	724,3	962,9	9	0,1	0,0
0,0	19,7	443,8	1199,7	1663,2	23	1,8	0,0	0,0	1,6	244,0	879,8	1125,5	13	0,1	0,0
0,0	1,4	403,9	1102,2	1507,5	20	0,0	0,0	0,0	2,8	259,9	858,0	1120,7	12	0,0	0,0
0,0	10,5	441,5	1140,5	1592,6	21	0,0	0,0	0,0	20,4	226,6	778,2	1025,3	10	0,0	0,0
0,0	20,1	456,0	1145,6	1621,8	21	1,4	0,0	0,0	0,0	237,4	852,7	1090,1	11	0,0	0,0
0,0	9,3	373,4	1093,5	1476,2	18	1,7	0,0	0,0	10,1	205,5	723,9	939,5	10	0,6	0,0
0,0	0,0	435,1	1178,6	1613,7	21	0,0	0,0	10,0	1,5	230,3	796,0	1037,8	10	0,8	0,0
0,0	36,8	402,6	1068,3	1507,7	20	0,1	0,0	0,0	0,0	256,0	788,6	1044,5	12	0,2	0,0
0,0	0,0	433,8	1139,4	1573,2	21	1,6	0,0	0,0	11,7	287,5	767,6	1066,9	11	0,0	0,0
0,0	84,9	391,8	1079,4	1556,1	19	6,8	0,0	8,0	58,8	231,9	857,6	1156,4	13	1,7	0,0
0,0	3,0	471,7	1155,4	1630,0	22	1,6	0,0	0,0	39,7	222,2	781,5	1043,4	12	1,0	0,0
0,0	0,8	436,0	1142,3	1579,1	20	1,9	0,0	5,0	3,4	219,8	766,6	994,7	8	0,4	0,0
0,0	3,4	419,1	1095,0	1517,5	20	0,2	0,0	0,0	20,3	220,7	699,8	940,8	9	1,0	0,0
0,0	11,2	450,7	1189,5	1651,4	22	2,6	0,0	21,0	30,6	239,6	851,7	1143,0	12	2,4	0,0
0,0	34,6	426,2	1090,1	1550,9	19	1,0	0,0	0,0	4,9	238,0	813,0	1056,0	12	1,6	0,0
0,0	26,8	440,7	1134,6	1602,2	22	1,8	0,0	0,0	2,2	235,7	790,5	1028,4	10	1,8	0,0
0,0	0,0	464,0	1131,1	1595,1	21	2,3	0,0	0,0	5,8	244,7	791,4	1041,8	11	0,7	0,0
0,0	13,3	376,4	1102,2	1491,9	18	4,6	0,0	0,0	2,2	209,7	728,0	939,9	10	1,7	0,0
0,0	0,0	451,0	1166,9	1617,9	20	1,7	0,0	0,0	4,7	225,1	811,6	1041,4	11	1,0	0,0
0,0	117,8	483,3	1246,2	1847,2	25	0,0	0,0	0,0	15,2	271,0	843,5	1129,7	13	0,8	0,0
0,0	0,0	446,3	1177,6	1623,9	21	1,9	0,0	0,0	0,0	247,6	833,2	1080,8	10	1,1	0,0
0,0	149,2	417,3	1156,4	1722,9	22	9,7	0,0	0,8	59,2	244,6	902,3	1206,9	14	3,7	0,0
0,0	25,2	484,8	1159,5	1669,4	22	6,9	0,0	0,0	28,5	222,3	840,0	1090,8	12	2,8	0,0
0,0	89,6	495,1	1181,5	1766,1	23	3,7	0,0	0,0	4,9	212,2	804,5	1021,6	10	3,0	0,0
0,0	0,0	452,1	1087,6	1539,7	20	1,0	0,0	0,0	30,8	215,6	782,3	1028,8	10	3,0	0,0
0,0	30,1	469,5	1167,6	1667,2	23	8,5	0,0	0,0	47,4	244,6	878,1	1170,1	12	4,1	0,0
0,0	13,4	444,1	1132,3	1589,8	21	3,1	0,0	0,0	18,4	244,9	823,5	1086,9	12	2,3	0,0
0,0	17,1	474,8	1161,1	1652,9	21	4,5	0,0	0,0	5,2	222,7	793,5	1021,4	10	4,3	0,0
0,0	6,0	461,0	1136,0	1603,0	20	1,3	0,0	0,0	15,7	233,0	773,5	1022,2	10	4,1	0,0
0,0	17,3	385,9	1109,8	1513,1	21	4,8	0,0	0,2	4,4	223,8	823,5	1051,9	10	4,1	0,0
0,0	0,0	472,7	1184,1	1656,8	23	11,7	0,0	0,0	12,5	234,0	823,5	1070,0	10	3,7	0,0
0,0	0,0	422,2	1096,8	1519,0	20	4,2	0,0	0,0	0,0	272,8	841,4	1114,3	11	2,4	0,0
0,0	0,0	460,0	1145,3	1605,3	21	6,7	0,0	0,0	0,0	244,1	807,4	1051,5	11	3,6	0,0
0,0	23,3	439,2	1141,0	1603,4	21	2,8	0,0	1,8	15,2	234,9	807,5	1059,3	11	1,6	0,0

Fonte: Elaborada pelo autor.

Observe que, com a estratégia CDR, é possível obter soluções factíveis para todas as instâncias, porém as soluções não apresentam boa qualidade se comparadas às soluções obtidas com as estratégias EI e RCD, principalmente no Cenário 2, pois apresentam valores da função objetivo bem maiores. De certa forma, isso mostra a importância que as decisões de consolidação

têm para a obtenção de um bom plano de distribuição e confirma a relevância da integração dos problemas de programação do *cross-dock* e de roteamento de veículos obter plano de distribuição mais eficiente.

Tabela 41 – Informações sobre as soluções obtidas com a estratégia CDR - 40 clientes.

Cenário 1								Cenário 2							
AN	AT	MS	TV	VF	NV	GAP		AN	AT	MS	TV	VF	NV	GAP	
						CD	R							CD	R
16,3	78,5	429,3	1861,9	2385,9	35	4,0	0,0	0,0	52,9	257,0	1408,6	1718,5	24	1,9	0,0
0,0	93,9	487,0	1826,7	2407,6	34	4,9	0,0	0,2	78,1	251,3	1287,7	1617,4	19	1,2	0,0
0,0	3,7	450,4	1705,2	2159,3	30	3,3	0,0	0,0	20,9	274,8	1400,3	1696,0	19	2,5	0,0
0,0	0,0	469,2	1818,5	2287,7	33	3,0	0,0	0,0	18,9	284,3	1282,4	1585,6	18	1,1	0,0
0,0	8,0	474,2	1978,1	2460,3	38	5,1	0,0	7,0	16,9	253,8	1339,9	1617,6	21	1,8	0,0
0,0	41,0	495,6	1872,9	2409,5	35	6,2	0,0	0,0	63,9	243,8	1300,6	1608,2	18	1,5	0,0
0,0	77,7	432,9	1706,0	2216,5	30	0,0	0,0	8,4	49,8	242,7	1280,5	1581,3	15	1,9	1,2
0,0	6,3	424,6	1770,6	2201,6	30	2,1	0,0	0,0	74,5	294,7	1463,6	1832,8	21	0,0	12,0
0,0	22,4	484,6	1861,0	2368,0	34	3,2	0,0	0,0	32,8	243,8	1333,4	1610,1	19	2,0	3,3
0,0	7,6	419,6	1704,8	2131,9	30	3,2	0,0	0,0	8,5	256,9	1253,9	1519,3	17	2,5	0,7
0,0	0,0	451,9	1822,5	2274,4	31	2,1	0,0	0,0	7,8	222,6	1306,3	1536,7	17	2,6	1,6
0,0	0,0	461,4	1735,6	2197,0	31	0,5	0,0	0,0	0,0	254,8	1305,2	1560,0	14	0,8	2,6
0,0	164,0	449,4	1737,1	2350,5	33	5,9	0,0	4,6	68,3	263,1	1498,7	1834,6	25	3,8	0,0
0,0	171,5	500,2	1885,1	2556,7	36	8,9	0,0	0,0	117,1	254,5	1339,2	1710,8	20	3,4	4,3
0,0	25,6	470,7	1833,6	2329,9	33	3,7	0,0	0,0	31,5	288,4	1379,4	1699,3	20	2,3	2,3
0,0	0,0	494,3	1842,8	2337,2	31	3,2	0,0	0,0	28,2	281,9	1301,0	1611,1	17	3,3	0,5
0,0	118,6	481,6	1935,8	2535,9	35	9,0	0,0	0,0	59,8	247,5	1306,5	1613,8	19	3,4	0,0
0,0	26,9	502,6	1897,4	2426,9	35	6,5	0,0	0,0	69,9	251,2	1339,4	1660,6	18	1,5	4,8
0,0	22,4	411,4	1711,8	2145,7	30	2,9	0,0	0,0	25,7	254,1	1268,7	1548,4	17	2,0	0,0
0,0	15,2	443,8	1722,4	2181,4	28	2,4	0,0	0,0	6,4	227,6	1233,8	1467,8	15	2,2	2,5
0,0	61,0	501,3	1796,3	2358,7	34	3,9	0,0	0,0	19,2	253,7	1336,3	1609,1	19	1,9	0,0
0,0	29,4	440,1	1702,4	2171,9	30	4,4	0,0	0,0	6,6	253,6	1332,9	1593,1	19	2,6	0,0
0,0	3,5	482,5	1815,7	2301,8	32	8,9	0,0	0,0	8,5	227,4	1219,4	1455,3	16	2,6	3,8
0,0	0,0	467,2	1788,2	2255,3	32	5,4	0,0	0,0	0,0	259,0	1214,1	1473,0	17	2,9	0,0
7,5	267,7	454,2	1831,6	2561,0	35	14,6	0,0	7,7	107,1	273,7	1450,9	1839,4	24	6,7	0,0
2,0	249,4	514,4	1883,0	2648,8	36	14,7	0,0	2,9	60,4	249,2	1426,8	1739,3	22	4,0	4,2
0,0	73,7	481,8	1763,2	2318,7	32	3,0	0,0	0,0	50,6	273,4	1348,4	1672,4	19	3,2	3,0
0,0	0,0	498,7	1797,8	2296,6	33	9,3	0,0	0,0	3,2	271,8	1337,7	1612,6	16	3,2	3,7
0,0	386,3	516,8	1826,5	2729,6	33	18,0	0,0	0,0	102,4	252,9	1392,9	1748,3	22	3,1	0,0
0,0	104,6	526,5	1960,3	2591,4	37	11,5	0,0	0,0	77,1	246,9	1276,4	1600,4	19	4,6	6,7
0,0	23,5	440,9	1660,3	2124,6	30	6,9	0,0	0,0	17,7	239,7	1328,1	1585,5	17	4,1	3,9
0,0	13,4	459,2	1623,4	2096,0	30	3,4	0,0	0,0	4,5	228,7	1270,2	1503,5	15	2,8	4,0
0,0	343,0	519,1	1915,9	2778,0	36	12,5	0,0	0,0	31,1	258,9	1371,6	1661,6	18	5,0	1,8
0,0	25,9	438,7	1649,4	2114,0	30	4,9	0,0	0,0	16,8	252,4	1288,7	1557,8	17	5,3	1,3
0,0	28,5	481,5	1793,3	2303,3	33	7,6	0,0	2,5	14,7	218,5	1261,6	1497,3	15	3,1	0,0
0,0	12,8	489,4	1754,5	2256,7	31	8,5	0,0	0,0	5,3	247,6	1197,4	1450,2	15	4,2	2,8
0,7	69,6	470,7	1799,8	2340,8	33	6,1	0,0	0,9	37,7	254,3	1324,5	1617,5	18	2,8	2,0

Fonte: Elaborada pelo autor.

A.3.4 Estratégia de entrega direta

Nas Tabelas 42, 43 e 44, reportamos as informações sobre as soluções obtidas com a estratégia DD para os conjuntos de instâncias com 15, 25 e 40 clientes, respectivamente.

Tabela 42 – Informações sobre as soluções obtidas com a estratégia DD - 15 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
0,0	209,3	301,3	679,0	1189,7	15	0,0	0,0	29,3	176,7	679,0	885,0	15	0,0
0,0	53,4	403,7	679,0	1136,2	15	0,0	0,0	13,3	171,1	679,0	863,4	15	0,0
0,0	3,9	374,7	679,0	1057,6	15	0,0	0,0	0,0	187,2	679,0	866,2	15	0,0
0,0	8,2	357,3	679,0	1044,5	15	0,0	0,0	0,0	182,4	679,0	861,4	15	0,0
0,0	81,8	384,0	679,0	1144,8	15	0,0	0,0	34,0	209,6	679,0	922,7	15	0,0
0,0	33,1	359,1	679,0	1071,2	15	0,0	0,0	0,0	180,8	679,0	859,9	15	0,0
0,0	78,2	390,7	679,0	1147,9	15	0,0	0,0	0,0	166,4	679,0	845,4	15	0,0
0,0	0,0	281,1	679,0	960,2	15	0,0	0,0	0,0	173,2	679,0	852,2	15	0,0
0,0	1,3	300,3	679,0	980,6	15	0,0	0,0	0,6	188,2	679,0	867,8	15	0,0
0,0	29,7	397,3	679,0	1106,1	15	0,0	0,0	0,0	173,5	679,0	852,5	15	0,0
0,0	13,4	383,9	679,0	1076,4	15	0,0	0,0	0,0	208,1	679,0	887,1	15	0,0
0,0	0,0	343,8	679,0	1022,9	15	0,0	0,0	0,0	184,5	679,0	863,5	15	0,0
0,0	46,6	266,9	679,0	992,6	15	0,0	0,0	45,1	198,3	679,0	922,4	15	0,0
0,0	99,4	423,0	679,0	1201,4	15	0,0	0,0	0,0	165,3	679,0	844,4	15	0,0
0,0	7,9	386,3	679,0	1073,3	15	0,0	0,0	0,0	182,2	679,0	861,3	15	1,5
0,0	0,0	321,6	679,0	1000,6	15	0,0	0,0	0,0	198,2	679,0	877,2	15	0,0
0,0	12,3	385,9	679,0	1077,3	15	0,0	0,0	63,8	218,4	679,0	961,3	15	0,0
0,0	13,1	374,1	679,0	1066,3	15	0,0	0,0	2,2	205,8	679,0	887,0	15	0,0
0,0	43,4	384,4	679,0	1106,8	15	0,0	0,0	0,0	184,4	679,0	863,4	15	0,0
0,0	0,0	281,6	679,0	960,6	15	0,0	0,0	0,0	194,7	679,0	873,7	15	0,4
0,0	0,0	304,3	679,0	983,4	15	0,0	0,0	178,1	265,9	679,0	1123,1	15	0,0
0,0	55,5	379,7	679,0	1114,2	15	0,0	0,0	0,0	161,7	679,0	840,7	15	0,0
0,0	24,4	392,4	679,0	1095,9	15	0,0	0,0	0,0	214,0	679,0	893,0	15	0,0
0,0	0,0	402,9	679,0	1081,9	15	0,0	0,0	0,0	205,6	679,0	884,6	15	0,0
0,0	126,8	287,8	679,0	1093,6	15	0,0	0,0	16,3	185,3	679,0	880,6	15	0,0
0,0	93,4	423,0	679,0	1195,5	15	0,0	0,0	1,4	174,3	679,0	854,7	15	0,9
0,0	0,0	387,7	679,0	1066,7	15	0,0	0,0	0,0	185,8	679,0	864,8	15	0,0
0,0	0,0	332,6	679,0	1011,6	15	0,0	0,0	1,9	190,1	679,0	871,1	15	0,4
0,0	262,3	422,0	679,0	1363,4	15	0,0	0,0	0,0	204,6	679,0	883,6	15	0,1
0,0	144,6	405,0	679,0	1228,7	15	0,0	0,0	0,0	193,8	679,0	872,9	15	1,6
0,0	21,9	402,2	679,0	1103,1	15	0,0	0,0	0,0	174,2	679,0	853,2	15	0,9
0,0	12,0	301,6	679,0	992,6	15	1,4	0,0	0,0	176,0	679,0	855,0	15	0,5
0,0	9,0	316,8	679,0	1004,9	15	0,0	0,0	0,0	209,0	679,0	888,0	15	0,0
0,0	0,0	378,9	679,0	1057,9	15	0,0	0,0	0,0	189,1	679,0	868,1	15	1,2
0,0	0,0	402,2	679,0	1081,3	15	1,2	0,0	0,0	211,8	679,0	890,8	15	0,0
0,0	0,0	392,2	679,0	1071,2	15	0,0	0,0	0,0	195,3	679,0	874,4	15	0,4
0,0	41,2	362,0	679,0	1082,3	15	0,1	0,0	10,7	191,3	679,0	881,0	15	0,2

Fonte: Elaborada pelo autor.

No caso da estratégia DD, em que são admitidas apenas entregas direta, as instâncias são resolvidas pelo *solver* e utilizando o Modelo M2 com as variáveis x_{0i} e $x_{i,N+1}$ fixadas no valor 1, ou seja, uma rota para cada cliente. Além disso, as restrições redundantes são eliminadas. O tempo de solução é limitado a 7200 segundos.

Conforme os números apresentados nas tabelas, temos que o número de veículos de entrega (NV) é igual ao número de clientes (N) em todas as soluções, pois são permitidas apenas

entregas diretas. Além disso, observe que o tempo total de viagem é o mesmo em todas as soluções. Isso acontece porque as instâncias de Solomon (SOLOMON, 1987), utilizadas para gerar as instância do problema estudado, apenas se diferenciam uma das outras pelas janelas de tempo e pelas demandas dos clientes.

Tabela 43 – Informações sobre as soluções obtidas com a estratégia DD - 25 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
0,0	73,2	392,6	1246,2	1712,0	25	0,0	0,0	0,0	209,3	1246,2	1455,5	25	0,2
0,0	4,9	434,9	1246,2	1685,9	25	0,2	0,0	0,0	188,2	1246,2	1434,4	25	0,3
0,0	0,1	423,7	1246,2	1670,0	25	1,4	0,0	0,0	177,2	1246,2	1423,4	25	0,0
0,0	0,0	407,1	1246,2	1653,3	25	0,4	0,0	0,0	184,9	1246,2	1431,1	25	0,1
0,0	12,3	435,9	1246,2	1694,5	25	1,8	0,0	0,0	217,7	1246,2	1463,9	25	0,1
0,0	1,4	387,6	1246,2	1635,2	25	0,0	0,0	0,0	217,8	1246,2	1463,9	25	0,0
0,0	3,1	423,8	1246,2	1673,0	25	0,0	0,0	0,0	188,6	1246,2	1434,8	25	0,0
0,0	0,0	448,2	1246,2	1694,4	25	1,4	0,0	0,0	211,9	1246,2	1458,1	25	0,0
0,0	0,0	345,1	1246,2	1591,3	25	1,7	0,0	0,0	177,8	1246,2	1424,0	25	0,6
0,0	0,0	417,7	1246,2	1663,9	25	0,0	0,0	0,0	208,5	1246,2	1454,7	25	0,8
0,0	0,1	387,0	1246,2	1633,4	25	0,1	0,0	0,0	229,9	1246,2	1476,1	25	0,2
0,0	0,0	421,8	1246,2	1668,0	25	1,6	0,0	0,0	247,9	1246,2	1494,1	25	0,0
0,0	24,3	375,8	1246,2	1646,3	25	6,8	0,0	3,7	218,4	1246,2	1468,3	25	1,7
0,0	0,0	457,0	1246,2	1703,1	25	1,6	0,0	0,0	192,6	1246,2	1438,8	25	1,0
0,0	0,0	416,0	1246,2	1662,2	25	1,9	0,0	0,0	180,9	1246,2	1427,1	25	0,4
0,0	0,0	395,3	1246,2	1641,5	25	0,2	0,0	0,0	189,7	1246,2	1435,9	25	1,0
0,0	0,0	440,8	1246,2	1687,0	25	2,6	0,0	0,0	220,1	1246,2	1466,3	25	2,4
0,0	0,0	405,2	1246,2	1651,4	25	1,0	0,0	0,0	219,6	1246,2	1465,8	25	1,6
0,0	0,0	432,7	1246,2	1678,9	25	1,8	0,0	0,0	200,2	1246,2	1446,4	25	1,8
0,0	0,0	438,2	1246,2	1684,4	25	2,3	0,0	0,0	202,7	1246,2	1448,9	25	0,7
0,0	0,0	354,9	1246,2	1601,1	25	4,6	0,0	0,0	181,9	1246,2	1428,1	25	1,7
0,0	0,0	426,7	1246,2	1672,9	25	1,7	0,0	0,0	201,8	1246,2	1448,0	25	1,0
0,0	117,8	483,3	1246,2	1847,2	25	0,0	0,0	0,0	237,7	1246,2	1483,9	25	0,8
0,0	0,0	430,8	1246,2	1677,0	25	1,9	0,0	0,0	215,1	1246,2	1461,3	25	1,1
0,0	95,9	409,9	1246,2	1752,0	25	9,7	0,0	6,4	226,6	1246,2	1479,2	25	3,7
0,0	18,2	475,8	1246,2	1740,1	25	6,9	0,0	1,6	199,8	1246,2	1447,6	25	2,8
0,0	72,2	495,1	1246,2	1813,5	25	3,7	0,0	0,0	191,0	1246,2	1437,2	25	3,0
0,0	0,0	438,3	1246,2	1684,5	25	1,0	0,0	0,0	201,6	1246,2	1447,8	25	3,0
0,0	15,1	469,5	1246,2	1730,8	25	8,5	0,0	0,0	234,2	1246,2	1480,4	25	4,1
0,0	0,0	435,3	1246,2	1681,5	25	3,1	0,0	0,0	224,6	1246,2	1470,8	25	2,3
0,0	0,0	448,6	1246,2	1694,8	25	4,5	0,0	0,0	210,6	1246,2	1456,8	25	4,3
0,0	6,0	448,2	1246,2	1700,4	25	1,3	0,0	0,0	215,0	1246,2	1461,2	25	4,1
0,0	0,0	376,1	1246,2	1622,3	25	4,8	0,0	0,0	194,8	1246,2	1441,0	25	4,1
0,0	0,0	470,3	1246,2	1716,5	25	11,7	0,0	0,0	214,5	1246,2	1460,7	25	3,7
0,0	0,0	412,8	1246,2	1659,0	25	4,2	0,0	0,0	245,1	1246,2	1491,3	25	2,4
0,0	0,0	459,6	1246,2	1705,8	25	6,7	0,0	0,0	229,8	1246,2	1476,0	25	3,6
0,0	12,3	425,6	1246,2	1684,1	25	2,8	0,0	0,3	208,6	1246,2	1455,1	25	1,6

Fonte: Elaborada pelo autor.

Independente do cenário e do número de clientes, a estratégia DD fornece as piores soluções para a maioria dos casos. Essa desvantagem é principalmente devido ao alto tempo de viagem apresentado pelas soluções quando comparado ao tempo de viagem das soluções obtidas pelas outras estratégia. Novamente, isso mostra a relevância de permitir o roteamento, ou seja, permitir a consolidação de cargas no sistema distribuição considerado.

Tabela 44 – Informações sobre as soluções obtidas com a estratégia DD - 40 clientes.

Cenário 1							Cenário 2						
AN	AT	MS	TV	VF	NV	GAP	AN	AT	MS	TV	VF	NV	GAP
0,0	42,4	419,8	2008,9	2471,1	40	4,0	0,0	0,0	245,0	2008,9	2253,9	40	1,9
0,0	40,5	487,0	2008,9	2536,4	40	4,9	0,0	0,0	221,6	2008,9	2230,4	40	1,2
0,0	0,0	420,8	2008,9	2429,7	40	3,3	0,0	0,0	236,5	2008,9	2245,4	40	2,5
0,0	0,0	454,3	2008,9	2463,2	40	3,0	0,0	0,0	241,9	2008,9	2250,8	40	1,1
0,0	0,0	455,4	2008,9	2495,6	40	5,1	0,0	0,0	222,1	2008,9	2262,4	40	1,8
0,0	14,5	487,0	2008,9	2510,3	40	6,2	0,0	0,0	209,8	2008,9	2218,7	40	1,5
0,0	58,3	404,9	2008,9	2472,0	40	0,0	0,0	0,0	212,6	2008,9	2221,5	40	1,9
0,0	0,0	405,1	2008,9	2414,0	40	2,1	0,0	23,0	268,9	2008,9	2300,7	40	0,0
0,0	0,0	469,2	2008,9	2478,1	40	3,2	0,0	0,0	221,4	2008,9	2230,3	40	2,0
0,0	0,0	399,2	2008,9	2408,1	40	3,2	0,0	0,0	225,9	2008,9	2234,8	40	2,5
0,0	0,0	434,4	2008,9	2443,3	40	2,1	0,0	0,0	193,5	2008,9	2202,4	40	2,6
0,0	0,0	439,5	2008,9	2448,4	40	0,5	0,0	0,0	220,4	2008,9	2229,3	40	0,8
0,0	85,0	450,8	2008,9	2544,6	40	5,9	0,0	0,0	253,5	2008,9	2262,4	40	3,8
0,0	118,1	489,3	2008,9	2616,3	40	8,9	0,0	0,3	234,5	2008,9	2243,6	40	3,4
0,0	24,3	470,0	2008,9	2503,1	40	3,7	0,0	0,0	260,4	2008,9	2269,2	40	2,3
0,0	0,0	479,2	2008,9	2488,1	40	3,2	0,0	0,0	244,8	2008,9	2253,6	40	3,3
0,0	102,1	481,6	2008,9	2623,9	40	9,0	0,0	0,0	229,8	2008,9	2270,0	40	3,4
0,0	23,7	502,6	2008,9	2535,2	40	6,5	0,0	0,0	230,5	2008,9	2239,4	40	1,5
0,0	0,0	411,4	2008,9	2420,3	40	2,9	0,0	0,0	221,0	2008,9	2229,8	40	2,0
0,0	0,0	418,3	2008,9	2427,1	40	2,4	0,0	0,0	206,1	2008,9	2214,9	40	2,2
0,0	3,3	480,1	2008,9	2492,2	40	3,9	0,0	0,0	231,8	2008,9	2240,7	40	1,9
0,0	0,0	412,5	2008,9	2421,4	40	4,4	0,0	0,0	230,7	2008,9	2239,5	40	2,6
0,0	0,0	478,3	2008,9	2487,2	40	8,9	0,0	0,0	200,6	2008,9	2209,5	40	2,6
0,0	0,0	455,8	2008,9	2464,7	40	5,4	0,0	0,0	221,2	2008,9	2230,1	40	2,9
0,0	211,4	456,2	2008,9	2676,5	40	14,6	0,0	28,6	276,5	2008,9	2313,9	40	6,7
0,0	198,2	505,6	2008,9	2712,7	40	14,7	0,0	0,0	238,4	2008,9	2247,3	40	4,0
0,0	66,7	469,8	2008,9	2545,4	40	3,0	0,0	0,0	258,1	2008,9	2266,9	40	3,2
0,0	0,0	500,7	2008,9	2509,6	40	9,3	0,0	0,0	255,4	2008,9	2264,2	40	3,2
0,0	309,0	513,8	2008,9	2863,0	40	18,0	0,0	0,0	239,7	2008,9	2279,9	40	3,1
0,0	89,9	526,5	2008,9	2625,2	40	11,5	0,0	0,0	241,4	2008,9	2250,3	40	4,6
0,0	5,4	430,9	2008,9	2445,2	40	6,9	0,0	0,0	232,8	2008,9	2241,7	40	4,1
0,0	0,0	444,9	2008,9	2453,8	40	3,4	0,0	0,0	208,2	2008,9	2217,1	40	2,8
0,0	297,6	518,4	2008,9	2824,9	40	12,5	0,0	0,0	251,1	2008,9	2260,0	40	5,0
0,0	0,0	437,6	2008,9	2446,5	40	4,9	0,0	0,0	245,6	2008,9	2254,5	40	5,3
0,0	4,1	474,4	2008,9	2487,4	40	7,6	0,0	0,0	210,2	2008,9	2219,1	40	3,1
0,0	0,0	475,5	2008,9	2484,4	40	8,5	0,0	0,0	229,9	2008,9	2238,8	40	4,2
0,0	47,1	460,0	2008,9	2518,6	40	6,1	0,0	1,4	232,6	2008,9	2245,5	40	2,8

Fonte: Elaborada pelo autor.

A.4 Gráfico de comparação entre as estratégias EI e RCD para as instâncias com 15 clientes do Cenário 1.

Nesta seção, apresentamos gráficos de comparação entre os tempos que compõem o valor da função objetivo das soluções obtidas com as estratégias EI e RCD para as instâncias com 15 clientes do Cenário 1. Esses gráficos são similares aos gráficos apresentados para as instâncias do Cenário 2 no Capítulo 3.

Na Figura 22, temos os gráficos de comparação entre os tempos operacionais, os tempos de viagem e os tempos de atraso das soluções obtidas com cada estratégia. Como apenas a solução da instância R101-1A apresentou tempo de antecipação igual a 3,3 com a estratégia EI, não incluímos o gráfico dos tempos de antecipação na figura. Além disso, como já mencionado

Figura 22 – Tempos de acordo com as estratégias EI e RCD - Cenário 1.

Fonte: Elaborada pelo autor.

no caso da estratégia RCD, não foi possível obter soluções factíveis para algumas instâncias e, por isso, não constam valores para essas instâncias nos gráficos.

De acordo com a análise apresentada no Capítulo 3, obtivemos soluções melhores para a maioria dos casos com a estratégia EI, porém com diferenças menos expressivas se comparado ao Cenário 2. Assim como acontece no Cenário 2, conforme pode ser observado na Figura 22, as diferenças entre as soluções obtidas com as duas estratégias estão principalmente no tempo de atraso apresentado. Para 22 instâncias, o tempo operacional obtido com a estratégia EI foi menor, em média, 1,9%, com diferenças entre 0,3% e 11%. No caso do tempo de viagem, para 25

instâncias, as soluções obtidas com a estratégia CDR apresentaram valores menores, em média, 3,1% com diferenças entre 0,03% e 11%. Já em relação ao tempo de atraso, para 28 instâncias, obtivemos tempo de atraso menor, em média, 59% com diferenças entre 13% e 100% para 27 instâncias e apenas uma com diferença de 0,2%.

SOLUÇÕES PARA AVALIAÇÃO DAS SOLUÇÕES HEURÍSTICAS

Neste apêndice, apresentamos informações sobre as soluções obtidas pelo *solver* ao resolver os conjuntos de instâncias com 15, 25 e 40 clientes utilizando o Modelo M2. Essas soluções são usadas para avaliar as soluções obtidas com os métodos de solução desenvolvidos nos Capítulos 4, 5 e 6.

O tempo de solução foi limitado a 1800 segundos para as instâncias com 15 e 25 clientes. No caso das instâncias com 40 clientes, o *solver* não obtém solução para a maioria das instâncias, principalmente do Cenário 2. Por isso, o tempo de solução é limitado a 7200 segundos na resolução das instâncias com 40 clientes.

Nas Tabelas 45 e 45, respectivamente para os Cenários 1 e 2, apresentamos o valor da função objetivo (VF) e o *gap* de otimalidade em porcentagem (GAP) das melhores soluções obtidas no tempo limite estabelecido. Além disso, também são reportados os tempos de solução em segundos (T). Lembrando que, no caso do *solver* de otimização Gurobi, o *gap* fornecido é a diferença percentual dos limitantes primal e dual em relação ao limitante primal. As tabelas estão divididas verticalmente em três partes que correspondem, de cima para baixo, às instâncias do Grupo A, B e C, respectivamente. E, na última linha, temos as médias dos valores de cada coluna.

De acordo com o que é apresentado nas tabelas, tivemos soluções com *gaps* nulos para 39 instâncias das 72 no caso com 15 clientes e apenas a instância R101-1A no caso das instâncias com 25 clientes. Observe que, mesmo dando um tempo maior de solução para as instâncias com 40 clientes, o *solver* não encontrou soluções para 9 instâncias do Cenário 2 e, por isso, não há valores na Tabela 45 para estas instâncias.

Tabela 45 – Informações sobre as soluções obtidas com o modelo M2 - Cenário 1.

Instância	15 clientes			25 clientes			40 clientes		
	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)
R101-1A	1090,0	0,0	24	1532,1	0,0	1217	2256,5	17,3	7200
R102-1A	1089,8	0,0	2	1500,0	0,7	1800	2335,8	10,4	7200
R103-1A	999,3	0,0	8	1488,2	2,4	1800	1989,1	6,1	7200
R104-1A	905,4	0,0	3	1397,3	2,8	1800	2081,2	5,2	7200
R105-1A	1119,3	0,0	890	1545,9	3,6	1800	2289,0	11,9	7200
R106-1A	982,2	0,0	3	1421,1	5,4	1800	2293,1	10,7	7200
R107-1A	1062,9	0,0	3	1470,9	0,5	1800	1924,7	1,3	7200
R108-1A	788,9	0,0	169	1475,2	2,3	1800	1877,8	3,4	7200
R109-1A	855,5	0,0	173	1287,4	5,6	1800	2129,0	5,2	7200
R110-1A	1058,4	0,0	1	1439,2	0,9	1800	1847,2	5,0	7200
R111-1A	1042,6	0,0	330	1406,4	6,4	1800	2034,1	2,7	7200
R112-1A	943,2	0,0	662	1530,5	2,9	1800	2079,6	2,4	7200
R101-1B	916,2	5,5	1800	1498,4	14,4	1800	2320,6	17,0	7200
R102-1B	1173,7	0,0	4	1532,6	4,4	1800	2418,9	14,6	7200
R103-1B	994,6	0,0	75	1481,4	4,9	1800	2185,5	12,1	7200
R104-1B	880,4	0,0	17	1425,7	10,8	1800	2121,8	6,0	7200
R105-1B	1048,4	0,0	4	1549,8	6,4	1800	2503,8	18,5	7200
R106-1B	982,6	0,0	6	1457,6	8,2	1800	2379,8	13,4	7200
R107-1B	1024,2	0,0	91	1480,2	4,9	1800	1963,5	10,2	7200
R108-1B	806,8	0,0	691	1465,7	5,7	1800	1904,0	4,4	7200
R109-1B	897,1	3,2	1800	1331,8	13,0	1800	2314,0	12,8	7200
R110-1B	1042,2	0,0	1001	1451,2	6,9	1800	1862,7	7,4	7200
R111-1B	1066,4	0,0	23	1753,2	6,3	1800	2154,2	14,0	7200
R112-1B	1032,8	0,0	70	1543,8	4,6	1800	2100,1	7,6	7200
R101-1C	1033,5	12,6	1800	1688,6	26,3	1800	2836,9	35,9	7200
R102-1C	1154,4	0,8	1800	1572,5	12,8	1800	2503,3	20,3	7200
R103-1C	1005,9	0,0	42	1687,6	7,9	1800	2268,1	13,4	7200
R104-1C	878,9	0,0	895	1436,5	2,5	1800	2264,9	17,9	7200
R105-1C	1367,0	13,4	1800	1636,9	16,3	1800	3064,6	34,7	7200
R106-1C	1173,7	0,0	537	1475,6	6,2	1800	2484,7	18,9	7200
R107-1C	1019,8	0,0	179	1491,7	6,7	1800	2169,6	22,8	7200
R108-1C	823,5	1,3	1800	1480,2	3,3	1800	1939,6	5,8	7200
R109-1C	915,1	0,0	72	1323,1	11,4	1800	2949,9	29,2	7200
R110-1C	985,3	2,9	1800	1511,3	14,6	1800	2007,6	12,9	7200
R111-1C	1035,8	2,7	1800	1452,5	9,3	1800	2187,6	13,8	7200
R112-1C	998,0	1,8	1800	1583,0	13,0	1800	2090,4	10,8	7200
Média	1005,4	1,2	616	1494,6	7,1	1784	2225,9	13	7200

Tabela 46 – Informações sobre as soluções obtidas com o modelo M2 - Cenário 2.

Instância	15 clientes			25 clientes			40 clientes		
	VF	GAP	T(s)	VF	GAP	T(s)	VF	GAP	T(s)
R101-2A	682,4	0,0	106	958,1	17,4	1800	1587,6	30,9	7200
R102-2A	610,5	18,7	1800	843,8	15,1	1800	-	-	-
R103-2A	633,4	5,2	1800	801,8	22,8	1800	1509,4	37,3	7200
R104-2A	635,6	11,4	1800	850,9	24,7	1800	1389,0	31,3	7200
R105-2A	761,9	0,6	1800	900,4	11,1	1800	1591,6	38,9	7200
R106-2A	594,4	2,7	1800	912,8	14,5	1800	1274,4	30,9	7200
R107-2A	570,7	7,0	1800	860,9	22,0	1800	1282,6	35,9	7200
R108-2A	579,6	3,9	1800	887,3	18,9	1800	-	-	-
R109-2A	639,5	0,0	1069	827,6	22,0	1800	1410,4	36,9	7200
R110-2A	552,2	0,0	1209	847,6	18,3	1800	1301,3	35,1	7200
R111-2A	646,3	0,0	800	921,5	11,1	1800	1386,4	45,6	7200
R112-2A	570,8	0,0	1201	911,4	14,3	1800	1266,6	32,8	7200
R101-2B	773,9	0,0	1183	1010,6	23,3	1800	1757,1	40,0	7200
R102-2B	572,7	11,7	1800	864,1	19,2	1800	1497,3	39,7	7200
R103-2B	567,8	2,2	1800	814,0	25,1	1800	1777,2	45,3	7200
R104-2B	656,7	11,8	1800	833,4	21,2	1800	1361,7	34,0	7200
R105-2B	800,0	0,0	1532	906,2	13,8	1800	1628,4	41,8	7200
R106-2B	678,5	0,0	1556	901,3	16,3	1800	1413,5	35,8	7200
R107-2B	591,2	0,0	779	853,1	21,2	1800	1489,0	44,3	7200
R108-2B	592,6	3,0	1800	850,2	19,8	1800	1262,5	39,4	7200
R109-2B	1075,1	31,0	1800	788,1	18,9	1800	1465,4	38,3	7200
R110-2B	537,8	9,8	1800	868,1	26,9	1800	1354,4	37,8	7200
R111-2B	666,5	13,6	1800	965,2	20,7	1800	1234,8	37,4	7200
R112-2B	595,2	13,5	1800	882,3	20,3	1800	1347,9	40,1	7200
R101-2C	680,2	8,2	1800	1013,0	24,5	1800	-	-	-
R102-2C	597,7	12,9	1800	882,7	19,6	1800	1939,8	53,8	7200
R103-2C	588,0	13,2	1800	814,3	24,1	1800	1572,3	39,2	7200
R104-2C	634,2	26,3	1800	884,6	24,7	1800	1408,4	35,7	7200
R105-2C	680,5	0,0	777	933,6	17,8	1800	-	0,0	-
R106-2C	595,1	4,1	1800	901,0	19,2	1800	1563,9	45,0	7200
R107-2C	567,1	5,9	1800	829,5	23,4	1800	-	-	-
R108-2C	561,2	1,3	1800	910,4	27,8	1800	-	-	-
R109-2C	684,6	11,7	1800	822,0	25,8	1800	-	0,0	-
R110-2C	558,2	0,0	967	894,1	26,4	1800	-	0,0	-
R111-2C	639,6	11,1	1800	941,2	18,7	1800	1283,9	40,1	7200
R112-2C	579,2	0,0	1084	865,5	21,5	1800	-	-	-
Média	637,5	6,7	1541	882,0	20,3	1800	1457,6	34,8	7200

SOLUÇÕES OBTIDAS COM A HEURÍSTICA CONSTRUTIVA

Neste apêndice, apresentamos informações sobre as soluções obtidas com a heurística construtiva que são comparadas com as soluções obtidas com o *solver* de otimização Gurobi utilizando o Modelo M2. Informações sobre estas soluções estão disponíveis no Apêndice B.

Foram propostas quatro versões para a heurística construtiva que resultam da combinação dos dois critérios de prioridade com os dois critérios de compatibilidade. A diferença entre os critérios nos dois casos, como explicado no Capítulo 4, é que eles podem ser dinâmicos ou estáticos. No caso estático, utilizamos a notação Pr1 para o critério de prioridade e Comp1 para o critério de compatibilidade. E, no caso dinâmico, denotamos por Pr2 e por Comp2 os critério de prioridade e de compatibilidade, respectivamente.

Nas Tabelas 47 e 49, para o Cenário 1 e para o Cenário 2 respectivamente, reportamos os valores da função objetivo das soluções obtidas pelas quatro versões da heurística construtiva para cada instância. Por conta do espaço, não são colocados os nomes das instâncias nas tabelas, mas, de cima para baixo, temos os valores referentes às soluções das instâncias R101-R112 do Grupo A, depois das instâncias R101-R112 do Grupo B e, por último, das instâncias R101-R112 do Grupo C. E, na última linha, temos as médias dos valores de cada coluna.

Em especial, no Cenário 1, grande parte das soluções são ineficazes por apresentarem um tempo extra para que as entregas sejam todas realizadas e os veículos retornem ao *cross-dock*. Neste caso, na Tabela 48, reportamos todos os valores do tempo extra apresentado em cada uma das soluções. A disposição das informações de acordo com as instâncias segue a mesma distribuição explicada no parágrafo anterior para as Tabelas 47 e 49. E, além disso, na última linha também temos as médias dos valores de cada coluna.

Tabela 47 – Valor da função objetivo das soluções obtidas com as quatro versões da heurística construtiva - Cenário 1.

15 clientes				25 clientes				40 clientes			
H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(1, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)
1795,0	1795,0	1758,8	1758,8	2861,5	2861,5	2814,5	2814,5	3351,1	3351,1	4973,6	4973,6
1163,9	1163,9	1332,8	1332,8	1864,3	1864,3	2477,1	2477,1	3645,0	3645,0	3053,6	3053,6
1310,3	1310,3	1157,4	1157,4	2025,9	2025,9	2002,1	1948,5	2863,7	2888,3	2578,6	2578,6
1048,6	1048,6	1020,0	1020,0	1694,2	1694,2	1742,9	1742,9	2279,6	2279,6	2684,2	2684,2
1516,9	1516,9	2367,3	2367,3	2197,7	2197,7	2691,7	2691,7	3245,3	3245,3	5832,1	5832,1
1222,8	1222,8	1218,6	1218,6	1698,2	1698,2	1732,6	1739,1	2915,6	2915,6	4146,6	4146,6
1336,3	1336,3	1338,0	1338,0	1642,9	1642,9	2302,1	2302,1	3002,2	3048,3	2773,0	2773,0
864,7	864,7	883,7	883,7	1815,4	1815,4	1665,3	1665,3	2153,2	2153,2	2680,1	2680,1
1198,5	1198,5	975,6	975,6	1728,6	1633,7	1802,9	1802,9	5265,6	5795,7	5133,0	5133,0
1726,9	1726,9	1641,9	1641,9	1940,6	2021,7	1822,7	2412,5	3724,0	3724,0	3996,4	3996,4
1082,4	1082,4	1139,9	1139,9	1760,1	1760,1	1889,7	1889,7	2447,9	2715,1	2771,4	2787,0
1059,5	1059,5	1010,4	1010,4	2250,2	2250,2	1834,0	1834,0	2579,8	2579,8	2915,0	2909,6
1092,4	1092,4	1078,1	1078,1	2006,1	1961,8	2491,8	2243,6	4937,5	4937,5	3511,2	3511,2
1680,6	1680,6	1433,7	1433,7	2342,7	2342,7	2830,2	2830,2	3810,3	3810,3	3838,5	3838,5
1046,9	1046,9	1075,1	1075,1	1655,3	1618,7	1712,6	1628,7	4185,2	4185,2	4913,0	4913,0
933,0	933,0	913,0	913,0	1559,6	1559,6	2008,7	2008,7	2710,6	2914,0	3154,6	3154,6
1503,2	1503,2	1641,1	1641,1	2236,9	2236,9	2216,2	2216,2	4385,4	4385,4	5151,8	5151,8
1017,8	1017,8	1195,9	1195,9	2765,3	2765,3	3865,3	3865,3	3845,4	3845,4	4002,7	4002,7
1219,3	1230,7	1218,0	1218,0	2187,5	2231,3	2066,5	2066,5	3209,1	3209,1	4743,9	4743,9
885,9	885,9	865,2	865,2	3655,0	3655,0	1977,3	1977,3	2382,6	2382,6	2321,2	2321,2
1181,4	1181,4	1136,7	1136,7	1661,3	1661,3	2514,3	2551,3	2960,6	2985,4	2807,6	2807,6
1265,8	1265,8	1300,1	1300,1	1835,6	1849,5	1787,0	1700,0	2584,6	2598,0	2504,3	2430,8
1163,1	1163,1	1232,5	1232,5	2672,3	2672,3	2443,6	2443,6	2793,1	2793,1	2747,4	2735,7
1518,4	1500,7	1639,1	1620,0	1923,0	1923,0	2589,1	2589,1	2452,6	2518,7	2735,4	2735,4
1199,1	1199,1	1197,1	1197,1	2275,9	2275,9	2281,5	2281,5	3887,3	3887,3	4129,5	4129,5
1365,9	1365,9	1248,6	1248,6	2461,8	2461,8	2419,8	2419,8	3447,5	3447,5	3216,3	3216,3
1068,8	1068,8	1123,1	1123,1	1840,5	1840,5	1900,8	1915,6	3113,6	3113,6	2471,8	2471,8
944,7	944,7	929,0	929,0	1784,9	1784,9	1743,6	1743,6	2629,6	2629,6	2396,2	2396,2
1693,2	1693,2	1697,0	1697,0	2446,3	2453,9	2291,4	2291,4	3589,2	3616,2	3418,9	3418,9
1398,2	1398,2	1372,4	1372,4	1703,8	1703,8	1713,5	1713,5	3714,5	3714,5	3495,0	3495,0
1188,1	1205,7	1094,1	1104,1	1760,2	1760,2	1713,2	1713,2	2490,0	2490,0	2925,2	2925,2
1036,1	1036,1	1031,3	1031,3	1670,0	1670,0	1643,6	1643,6	2150,1	2150,1	2144,4	2144,4
1113,0	1096,3	1288,8	1269,1	1744,7	1731,1	1606,9	1647,3	4435,1	4435,1	3807,1	3869,9
1075,9	1075,9	1081,7	1081,7	2074,5	2127,0	1970,4	1999,3	2677,1	2658,2	2743,9	2705,9
1073,7	1073,7	1079,5	1079,5	1586,2	1586,2	1656,1	1656,1	2650,3	2650,3	2459,1	2519,6
1190,2	1190,2	1161,9	1161,9	1871,4	1871,4	1850,4	1850,4	2640,1	2664,8	2864,7	2876,1
1227,2	1227,1	1246,6	1245,8	2033,3	2033,6	2113,1	2119,9	3198,7	3232,3	3390,0	3390,6

Tabela 48 – Tempo extra apresentado nas soluções obtidas com as quatro versões da heurística construtiva - Cenário 1.

15 clientes				25 clientes				40 clientes			
H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(1, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)
0,0	0,0	0,0	0,0	33,6	33,6	23,8	23,8	24,9	24,9	62,4	62,4
0,0	0,0	22,5	22,5	0,1	0,1	49,0	49,0	54,3	54,3	11,0	11,0
0,0	0,0	0,0	0,0	27,8	27,8	53,1	42,8	10,3	22,0	19,1	19,1
0,0	0,0	11,7	11,7	21,4	21,4	19,5	19,5	12,6	12,6	60,1	60,1
18,3	18,3	76,0	76,0	24,7	24,7	36,5	36,5	40,2	40,2	123,2	123,2
12,2	12,2	0,0	0,0	0,0	0,0	0,0	0,0	36,1	36,1	92,7	92,7
26,8	26,8	23,2	23,2	10,5	10,5	84,7	84,7	44,7	44,7	66,0	66,0
0,0	0,0	0,0	0,0	16,7	16,7	4,3	4,3	20,9	20,9	68,3	68,3
6,1	6,1	0,0	0,0	0,0	0,0	0,0	0,0	135,0	157,1	91,2	91,2
44,3	44,3	57,2	57,2	25,9	34,8	6,7	48,7	60,5	60,5	72,0	72,0
0,0	0,0	0,0	0,0	23,5	23,5	44,0	44,0	63,9	68,9	63,0	63,1
0,0	0,0	0,0	0,0	97,8	97,8	52,9	52,9	27,9	27,9	36,5	36,5
0,0	0,0	0,0	0,0	0,0	0,0	43,8	25,9	90,4	90,4	19,6	19,6
45,4	45,4	27,4	27,4	51,0	51,0	75,0	75,0	41,7	41,7	47,0	47,0
3,6	3,6	1,6	1,6	0,0	0,0	0,0	0,0	97,7	97,7	94,6	94,6
0,0	0,0	0,0	0,0	7,5	7,5	46,9	46,9	76,1	78,7	89,6	89,6
24,7	24,7	23,2	23,2	17,8	17,8	35,1	35,1	78,2	78,2	90,8	90,8
0,0	0,0	5,9	5,9	69,3	69,3	111,7	111,7	60,1	60,1	98,4	98,4
9,4	11,4	21,1	21,1	106,2	108,0	87,9	87,9	78,1	78,1	111,4	111,4
0,0	0,0	0,0	0,0	137,1	137,1	36,6	36,6	49,6	49,6	26,0	26,0
0,0	0,0	0,0	0,0	0,0	0,0	35,6	39,9	26,0	23,2	22,2	22,2
5,3	5,3	21,3	21,3	21,9	14,5	15,7	27,2	19,5	17,9	0,0	6,6
0,0	0,0	11,4	11,4	91,6	91,6	98,0	98,0	26,3	26,3	73,0	55,4
49,6	49,6	61,4	58,6	37,0	37,0	88,7	88,7	36,8	36,9	57,3	57,3
0,0	0,0	0,0	0,0	0,0	0,0	2,8	2,8	31,8	31,8	60,1	60,1
16,6	16,6	4,3	4,3	36,1	36,1	48,2	48,2	57,6	57,6	56,2	56,2
0,0	0,0	4,5	4,5	38,3	38,3	43,6	43,1	39,3	39,3	1,5	1,5
0,0	0,0	0,0	0,0	47,0	47,0	32,6	32,6	37,5	37,5	53,7	53,7
13,0	13,0	26,7	26,7	40,7	47,7	23,6	23,6	38,4	40,9	33,9	33,9
14,0	14,0	3,1	3,1	1,3	1,3	0,0	0,0	76,5	76,5	85,2	85,2
19,9	19,9	11,1	11,1	23,1	23,1	33,1	33,1	35,2	35,2	59,6	59,6
0,0	0,0	0,0	0,0	12,5	12,5	28,4	28,4	14,3	14,3	1,2	1,2
0,0	0,0	0,1	1,6	0,0	0,0	0,0	0,0	94,0	94,0	58,3	58,3
0,0	0,0	0,0	0,0	52,8	49,9	43,6	44,7	27,1	37,2	45,9	45,9
0,0	0,0	0,0	0,0	14,4	14,4	27,4	27,4	48,6	48,6	25,2	25,3
18,1	18,1	7,0	7,0	52,7	52,7	39,6	39,6	42,3	40,8	49,5	45,5
9,1	9,2	11,7	11,7	31,7	31,9	38,1	39,0	48,7	50,1	56,3	55,9

Tabela 49 – Valor da função objetivo das soluções obtidas com as quatro versões da heurística construtiva - Cenário 2.

15 clientes				25 clientes				40 clientes			
H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(1, 1)	H(2, 2)	H(1, 1)	H(1, 2)	H(2, 1)	H(2, 2)
922,3	922,3	871,2	871,2	1309,8	1309,8	1268,7	1268,7	2237,3	2237,3	2359,4	2380,9
725,2	716,7	802,6	802,6	1266,4	1172,7	1126,1	1095,5	2152,2	2162,6	2196,8	1876,3
916,4	916,4	799,1	799,1	1067,2	933,4	1041,1	1059,4	1608,2	1608,2	2052,1	2052,1
796,8	708,5	792,4	713,4	1021,3	939,1	1053,7	1002,7	1912,3	1912,3	1782,9	1783,7
998,3	998,3	983,1	983,1	1240,3	1160,3	1154,7	1243,5	2276,0	2250,4	2100,2	2110,1
742,7	767,3	744,8	744,8	1290,0	1260,8	1218,5	1148,3	1894,9	1812,6	2196,8	2177,1
761,8	761,8	723,9	723,9	1026,4	1033,9	968,8	968,8	1646,9	1619,4	1782,0	1764,7
672,4	712,9	672,4	733,3	1089,1	1071,1	1027,4	1031,6	2193,7	2172,3	1837,9	1702,8
841,0	838,1	908,0	908,0	981,4	1149,0	1019,6	1144,5	1987,2	2119,7	1900,6	2041,3
788,9	814,4	733,7	775,1	1216,8	1144,9	1051,4	1074,7	1520,6	1727,9	1499,1	1597,9
796,5	796,5	796,5	796,5	1316,0	1172,0	1227,2	1227,2	1561,6	1446,1	1326,8	1307,2
637,5	647,3	765,3	677,6	1133,2	1133,2	1161,5	1161,5	1450,6	1425,7	1521,2	1451,7
961,2	961,2	959,3	959,3	1532,8	1532,8	1879,8	1911,1	1943,4	1945,9	2377,7	2349,3
673,9	707,7	682,2	722,0	1186,6	1186,6	1140,1	1020,1	2073,8	2088,8	1986,3	1986,3
728,2	728,2	739,5	739,5	1107,1	1104,0	1046,9	1072,2	1673,2	1676,3	1844,8	1844,8
876,5	746,3	879,3	750,8	1010,3	1010,3	1069,5	1069,5	1553,6	1569,0	1502,3	1502,3
1017,3	1009,4	995,0	995,0	1133,1	1133,1	1174,3	1108,0	1759,8	1695,2	1930,7	1821,4
869,0	809,8	871,0	825,9	1217,9	1179,8	1371,1	1316,7	2272,4	2210,0	1730,4	1773,2
714,7	714,7	714,7	714,7	961,0	920,8	963,7	973,6	1516,3	1625,8	1603,2	1504,6
733,7	733,7	785,4	785,4	1026,7	1026,7	1021,8	1041,4	1349,7	1376,9	1369,0	1365,5
1468,8	1468,8	1492,8	1492,8	1030,3	970,0	1151,0	1021,3	1821,5	1745,2	1766,2	1534,8
725,5	674,2	734,7	749,3	1009,9	1004,3	1078,4	1021,2	1648,5	1527,0	1657,8	1591,6
734,2	734,2	812,7	812,7	1157,1	1142,4	1222,9	1189,8	1463,6	1360,8	1508,3	1514,5
691,0	761,9	731,0	762,3	1003,9	1003,9	1050,7	1040,9	1656,2	1375,1	1648,1	1318,9
754,9	754,9	818,8	818,8	1298,5	1306,5	1323,2	1323,2	2774,0	2805,4	2926,4	2926,4
653,9	657,6	646,5	655,8	1215,0	1170,1	1130,3	1130,3	1924,5	1870,7	1898,4	1832,2
750,6	640,8	651,6	651,6	943,0	943,0	1028,7	966,4	1878,9	1878,9	1847,6	1827,0
754,8	726,0	756,6	726,0	1100,9	1100,9	1148,0	1148,0	1462,3	1462,3	1452,7	1452,7
804,4	761,6	804,4	761,6	1279,4	1266,1	1238,6	1234,1	1770,1	1826,4	1726,7	1695,6
672,4	672,4	674,4	674,4	1129,7	1052,9	1142,5	1119,2	1879,3	1859,0	1849,1	1849,1
642,6	643,9	642,6	643,9	939,5	925,7	996,3	969,0	1568,2	1468,9	1537,9	1457,3
699,3	699,3	687,0	687,0	1055,1	1055,1	962,6	962,6	1244,6	1273,5	1259,0	1266,8
863,1	863,1	859,7	859,7	1117,5	1013,5	1095,4	1063,0	1683,3	1579,2	1667,5	1582,1
690,7	733,8	690,7	733,8	1018,7	967,7	993,3	921,6	1609,9	1442,3	1511,5	1493,3
768,0	768,0	735,3	735,3	1063,4	1122,2	1052,8	1102,0	1834,7	1661,0	1779,3	1641,0
722,3	722,3	656,2	656,2	938,5	931,2	1037,1	1037,1	1341,7	1406,5	1256,7	1347,7
793,6	786,0	794,8	790,1	1123,2	1098,6	1128,8	1116,4	1781,8	1756,2	1783,1	1742,3

SOLUÇÕES OBTIDAS COM A META-HEURÍSTICA

Neste apêndice, apresentamos informações sobre as soluções obtidas com a meta-heurística na Fase 1 dos experimentos do Capítulo 5. As soluções obtidas são comparadas com as soluções fornecidas pelo *solver* de otimização Gurobi utilizando o Modelo M2. Informações sobre estas soluções estão disponíveis no Apêndice B.

Nas Tabelas 50 e 51, respectivamente para o Cenário 1 e para o Cenário 2, reportamos o valor da função objetivo de cada solução e o tempo de processamento da meta-heurística para obtê-la. Por conta do espaço, não são colocados os nomes das instâncias nas tabelas, mas, de cima para baixo, temos os valores referentes às soluções das instâncias R101-R112 do Grupo A, depois das instâncias R101-R112 do Grupo B e, por último, das instâncias R101-R112 do Grupo C. E, na última linha, temos as médias dos valores de cada coluna.

Tabela 50 – Valor da função objetivo e tempo de resolução das soluções obtidas com as duas versões da FO-VND - Cenário 1.

FO-VND1						FO-VND2					
15 clientes		25 clientes		40 clientes		15 clientes		25 clientes		40 clientes	
VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)
1098,7	4	1547,1	114	2182,7	1791	1090,0	8	1544,5	126	2156,2	1851
1089,8	1	1542,3	29	2312,3	1802	1089,8	14	1546,0	199	2476,9	1819
1021,0	33	1503,3	403	1997,9	1333	999,7	23	1486,3	379	1981,4	1557
907,9	2	1410,0	182	2089,4	231	908,8	7	1421,6	384	2065,0	1115
1126,9	1	1572,3	157	2358,0	645	1125,5	7	1570,8	453	2394,5	1802
1039,2	1	1489,4	24	2303,0	974	984,4	3	1459,1	328	2295,1	1801
1084,7	4	1475,7	39	1947,4	1455	1062,9	4	1470,9	337	1951,8	1816
804,5	3	1487,5	246	1909,3	911	789,2	10	1480,7	578	1895,5	1847
886,1	9	1332,0	203	2211,9	1720	855,5	14	1272,4	816	2149,4	1802
1058,4	2	1457,8	17	1905,2	1801	1067,2	1	1438,3	120	1870,4	1804
1044,4	12	1415,2	37	2106,8	1117	1044,4	12	1399,0	435	2058,3	1816
966,4	106	1553,3	103	2080,8	867	943,2	84	1541,5	1303	2080,5	1802
926,3	41	1527,9	39	2265,8	1809	921,2	79	1513,6	900	2359,3	1809
1173,7	2	1584,9	324	2379,9	1143	1173,7	2	1568,1	186	2488,7	1738
994,6	19	1492,7	183	2051,2	1813	994,6	18	1486,1	624	2101,7	1820
887,1	2	1428,8	166	2099,1	1606	887,1	8	1412,8	958	2105,3	1359
1048,4	2	1604,5	356	2417,0	1623	1048,4	5	1545,7	649	2588,2	1801
982,9	2	1577,7	32	2290,5	1624	982,6	6	1456,1	550	2327,7	1850
1058,8	21	1505,7	494	1905,2	1325	1024,2	14	1474,6	1000	1968,7	1860
806,8	6	1490,9	489	1897,4	1507	812,5	14	1490,2	986	1927,4	1814
907,0	8	1350,7	83	2153,1	1812	899,7	30	1290,8	791	2188,8	1818
1056,3	9	1473,0	101	1845,8	1656	1051,5	13	1461,5	710	1912,3	1801
1067,0	2	1747,1	307	2083,7	1342	1066,4	5	1714,9	408	2089,5	1819
1039,9	5	1563,5	87	2093,3	1801	1035,6	7	1534,7	311	2097,7	1808
1043,2	31	1595,4	482	2443,0	1841	1026,1	163	1609,9	1044	2453,4	1817
1157,0	58	1652,1	383	2457,7	1547	1154,4	57	1627,4	1242	2561,8	1216
1005,9	7	1708,1	168	2103,8	1319	1005,9	13	1683,2	381	2120,6	1843
879,5	8	1439,6	471	2092,3	1805	878,9	19	1474,7	1112	2111,0	1840
1360,0	130	1665,4	297	2538,4	1821	1360,0	143	1625,4	771	2669,1	1236
1173,7	28	1460,9	407	2375,3	1370	1186,0	43	1459,6	822	2428,3	1817
1036,7	8	1523,5	306	1911,6	1581	1021,4	18	1500,1	1086	1975,0	1838
833,6	22	1480,3	273	1920,9	1753	823,5	19	1478,9	515	1960,2	1623
917,1	7	1307,5	683	2587,8	1534	915,1	27	1327,9	1140	2704,6	1806
988,7	36	1501,8	476	1854,4	1813	982,4	39	1559,1	930	1945,4	1457
1036,8	79	1448,6	117	2067,1	1820	1036,8	46	1448,0	1014	2140,6	1817
998,7	74	1568,3	467	2078,3	1287	998,0	112	1559,5	1011	2145,0	1817
1014,1	22	1513,5	243	2147,7	1478	1006,8	30	1498,2	683	2187,4	1729

Tabela 51 – Valor da função objetivo e tempo de resolução das soluções obtidas com as duas versões da FO-VND - Cenário 2.

FO-VND1						FO-VND2					
15 clientes		25 clientes		40 clientes		15 clientes		25 clientes		40 clientes	
VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)	VF	T (s)
686,8	15	933,2	121	1487,3	1829	682,4	25	941,9	1028	1612,4	1820
634,7	50	872,0	129	1334,6	1120	631,2	185	867,3	615	1507,3	1253
633,4	53	804,3	302	1257,0	1776	635,7	85	840,0	734	1343,6	1740
637,4	79	827,2	268	1294,1	1728	637,9	78	822,2	428	1343,6	1832
762,2	34	1041,9	120	1359,7	1801	768,1	117	892,0	914	1473,8	1806
640,0	44	942,0	307	1300,2	1380	594,4	136	904,2	415	1416,7	1174
611,6	75	818,9	417	1192,4	1166	573,1	89	814,6	548	1255,0	1208
579,6	47	880,0	230	1378,0	1809	579,6	51	867,6	800	1371,2	1805
641,1	66	768,0	265	1245,2	1701	644,8	79	792,9	304	1284,5	1496
552,2	58	841,7	264	1206,0	1323	552,2	64	850,3	627	1269,2	1379
646,3	34	918,3	94	1123,5	1243	646,3	19	918,2	925	1170,3	992
570,8	61	923,1	178	1125,9	1080	570,8	95	907,9	284	1170,8	1225
773,9	57	969,7	561	1472,8	1325	773,9	58	990,5	1111	1559,2	1853
607,3	26	858,9	376	1342,8	1626	607,3	164	852,3	528	1629,4	1438
567,8	22	813,9	248	1413,5	1558	567,8	36	811,5	722	1500,1	1257
672,0	107	814,7	431	1300,6	1203	661,8	73	860,6	458	1352,7	1523
800,0	76	914,1	222	1324,2	1801	800,2	115	917,6	495	1548,2	1165
680,8	66	890,4	341	1284,7	1619	678,5	106	877,4	944	1380,2	1240
591,2	60	806,0	252	1248,3	1306	591,2	84	818,5	324	1422,2	800
610,1	26	850,1	306	1069,0	1297	594,1	33	885,4	441	1151,4	1419
1084,7	38	794,0	239	1245,8	1413	1080,8	64	841,4	443	1261,2	1666
538,1	26	876,0	235	1161,3	1820	553,5	167	851,8	1131	1261,3	1124
654,4	32	945,8	206	1088,3	1331	654,4	167	921,0	1324	1197,5	810
624,6	33	878,7	386	1155,3	1123	595,2	283	869,2	565	1218,7	1346
684,5	42	1016,4	258	1626,4	1818	680,2	148	1033,5	983	1959,4	1800
590,6	41	873,6	369	1414,3	1677	597,7	252	881,9	1018	1608,6	1156
589,1	39	785,7	392	1350,1	1531	588,0	180	818,4	536	1442,1	1699
615,0	59	819,4	295	1263,0	1812	619,2	218	860,7	862	1319,6	1458
716,3	14	946,5	321	1311,0	1716	683,5	32	1016,2	533	1554,8	1257
595,1	14	893,8	456	1244,6	1801	595,1	35	899,2	802	1498,0	1300
571,4	69	834,0	289	1159,0	1843	567,1	148	838,3	1208	1319,2	864
566,2	47	837,8	402	1064,2	1474	561,2	74	841,6	854	1133,8	1236
685,6	108	796,5	360	1239,5	1802	684,8	93	850,3	563	1383,2	1324
561,9	17	854,0	239	1186,8	1257	558,2	138	844,4	619	1265,4	1297
684,0	85	937,7	207	1078,1	1803	641,8	134	965,4	781	1244,2	1166
584,4	27	865,5	172	1134,8	1713	584,4	62	865,5	419	1158,3	1781
645,7	48	873,4	285	1263,4	1545	639,9	108	878,7	702	1377,4	1381