

UNIVERSIDADE FEDERAL FLUMINENSE

MARCOS RAYLAN SOUSA MATOS

**Problema de Roteamento de Veículos voltado para  
redução de emissões de carbono**

NITERÓI

2018

UNIVERSIDADE FEDERAL FLUMINENSE

MARCOS RAYLAN SOUSA MATOS

# Problema de Roteamento de Veículos voltado para redução de emissões de carbono

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização

Orientador:

Yuri Abitbol de Menezes Frota

Co-orientador:

Luiz Satoru Ochi

NITERÓI

2018

Ficha catalográfica automática - SDC/BEE

S725p    Sousa Matos, Marcos Raylan  
          Problema de Roteamento de Veículos voltado para redução  
          de emissões de carbono / Marcos Raylan Sousa Matos; Yuri  
          Abitbol de Menezes Frota, orientador; Luiz Satoru Ochi,  
          coorientador. Niterói, 2018.  
          67 f.

          Dissertação (mestrado)-Universidade Federal Fluminense,  
          Niterói, 2018.

          1. Problema de roteamento de veículos. 2. Emissão de CO2.  
          3. Metaheurística híbrida. 4. Programação Inteira. 5.  
          Produção intelectual. I. Título II. Menezes Frota, Yuri  
          Abitbol de, orientador. III. Ochi, Luiz Satoru, coorientador.  
          IV. Universidade Federal Fluminense. Escola de Engenharia.

CDD -

Marcos Raylan Sousa Matos

Problema de Roteamento de Veículos voltado para redução de emissões de carbono

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Algoritmos e Otimização

Aprovada em fevereiro de 2018.

BANCA EXAMINADORA

---

Prof. Yuri Abitbol de Menezes Frota - Orientador, UFF

---

Prof. Luiz Satoru Ochi - Co-orientador, UFF

---

Prof. Warley Gramacho da Silva - UFT

---

Prof. Igor Machado Coelho - IME/UERJ

---

Prof. Simone de Lima Martins - UFF

---

Prof. Uevertton Souza - UFF

Niterói

2018

*I'm feeling supersonic, Give me gin and tonic (Supersonic - Oasis)*

# Agradecimentos

Mesmo as palavras sendo insuficientes para expressar a gratidão que tenho a todos que contribuirão durante esta etapa, a princípio gostaria de agradecer a minha família pelos conselhos, pela compreensão e até mesmo pelo apoio emocional que, de alguma forma, me fizeram acreditar que poderia alcançar os meus objetivos de vida.

Um agradecimento mais do que especial a minha mãe (Dona Silvana) por todo o seu amor e carinho. Gostaria de agradecer também a meu pai (Carlos) e meus irmãos (Alann e Aline) por todo apoio ao longo da minha vida, especialmente nos tempos de graduação. Um agradecimento especial aos amigos de longa data: Flanúbio, Áureo, Lucas, Luciano, Cássio, Rodrigo e a todos que estiveram a minha volta por tanto tempo. Fica aqui meus agradecimentos aos colegas de Niterói, Ítalo e Igor Barreto. Vale uma menção mais do que especial aos meus colegas de graduação Herinson e Sávio, a trupe mais fera da UFT.

Uma menção de honra a meus orientadores Yuri Abitbol e Luiz Satoru Ochi pelo apoio, suporte, direcionamento e incentivo na realização desse trabalho. Aos demais professores do IC/UFF e também aos membros da banca pelos comentários e contribuições que tornaram possível a concretização desse trabalho. Aos professores Puca Penna, Matheus Haddad e Marcos Melo pelas ideias e colaborações. Ao meu orientador da graduação, Prof. Warley Gramacho, por ter acreditado e me incentivado a trilhar o caminho da pós-graduação e pesquisa.

Gostaria de aproveitar esse momento também para agradecer ao Programa de Pós-graduação em Computação do Instituto de Computação da UFF pela oportunidade. Gostaria de agradecer pelo fornecimento dos recursos computacionais por parte do Laboratório de Algoritmos, Grafos e Otimização (LAGO - UFF) da UFF e o apoio fornecido pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

A todos, meu muito obrigado!

# Resumo

Esta dissertação aborda o *Heterogeneous Green Vehicle Routing and Scheduling Problem* (HGVRSP), uma variante do Problema de Roteamento de Veículos (PRV) com considerações ambientais. No HGVRSP, o objetivo é minimizar as emissões de Dióxido de Carbono ( $CO_2$ ) lançado pelos veículos utilizados no roteamento ao longo de vários períodos de tempo, no qual, certos períodos tendem a favorecer maiores emissões de  $CO_2$ . Nesse trabalho, o HGVRSP permite que os veículos parem no trajeto, dessa forma, um veículo tem permissão para percorrer um percurso em múltiplos períodos de tempo. Como método de resolução, duas abordagens são apresentadas. A primeira abordagem é baseada na meta-heurística *Multi-Start Iterated Local Search* (MS-ILS) e incorpora na etapa de busca local o procedimento de *Random Variable Neighborhood Descent* (RVND), sendo que a busca local reúne tanto estruturas de vizinhanças clássicas do PRV quanto vizinhanças específicas para explorar características do HGVRSP. A segunda abordagem apresentada utiliza um algoritmo exato que baseia-se em uma formulação matemática proposta para descrever o HGVRSP. Além disso, é apresentada uma abordagem híbrida que utiliza um método exato baseado no Problema de Cobertura de Conjuntos na fase de intensificação. Para mostrar a eficácia dos métodos, foram realizados experimentos em seis grupos de instâncias disponíveis na literatura, totalizando 120 instâncias. Os resultados obtidos são, em média, bastante competitivos quando comparados com o atual estado-da-arte.

**Palavras-chave:** Roteamento de Veículos, Logística Verde, Meta-heurísticas híbridas, Programação Inteira, Emissão de  $CO_2$ , Congestionamento de Tráfego.

# Abstract

This dissertation deals with the Heterogeneous Green Vehicle Routing and Scheduling Problem (HGVRSP), which is a variant of Vehicle Routing Problem (VRP) with environmental considerations. In the HGVRSP, the objective is to minimize emissions of carbon dioxide ( $CO_2$ ) issued by vehicles used in routing over several time periods, such that, some periods tend to favor higher  $CO_2$  emissions. In this paper, the HGVRSP allows vehicles to stop on the road, thus, a vehicle is allowed to traverse a route in multiple time periods. As resolution method, two approaches are presented. The first approach is based on Multi-Start Iterated Local Search (MS-ILS) metaheuristic and incorporates in the local search step the Random Variable Neighborhood Descent (RVND), the local search that includes classic neighborhoods structures of VRP, as well as those specific neighborhoods to explore HGVRSP features. The second approach applies an exact algorithm that is based on a mathematical formulation proposed to describe the HGVRSP. Furthermore, we propose a hybrid approach that uses a Set Covering (SC) formulation during an intensification phase. In order to validate our methods, we perform computational experiments on six instance groups of literature, in a total of 120 instances. The obtained results are competitive compared to the current state-of-art methods.

**Keywords:** Vehicle Routing, Green Logistics, Hybrid Metaheuristics, Integer Programming,  $CO_2$  Emission, Traffic Congestion.



# Lista de Figuras

1.1	Emissões de $CO_2$ por combustíveis fósseis, Fonte: [27]. . . . .	2
2.1	Representação de parte da rota $v$ na forma de <i>time-line</i> . . . . .	12
2.2	Solução com troca de períodos consecutivos $k$ e $k + 1$ . . . . .	13
2.3	Solução com parada entre os períodos $k$ e $k + 2$ . . . . .	14
2.4	Solução para uma instância com 15 clientes, 5 períodos, 3 veículos e emissão de 792.36 $kg/km$ de $CO_2$ . . . . .	15
4.1	Solução com <i>Split-Delivery</i> . . . . .	24
4.2	Agenda de acesso aos clientes. . . . .	25
4.3	Exemplo de Recombinação de Sequências. . . . .	32
4.4	Sequências Recombinadas para Execução de Movimentos. . . . .	32
4.5	Penalização da Janela de Tempo. . . . .	33
4.6	Vizinhanças Inter-Rotas [48]. . . . .	36
4.7	Vizinhanças Intrarrotas [48]. . . . .	37
4.8	Vizinhança inter-rota <i>RouteAddition</i> . . . . .	38
4.9	Vizinhança inter-rota <i>k-Split</i> . . . . .	38
5.1	Resultados para abordagem exata, instâncias com 10 vértices e 5 períodos. . . . .	44
5.2	Resultados para abordagem heurística, instâncias com 10 vértices e 5 períodos. . . . .	46

# Lista de Tabelas

2.1	Tipologia . . . . .	5
5.1	Comparação entre soluções com e sem <i>Split</i> para o grupo de instância $10 \times 5$ . . . . .	43
5.2	Resultados para abordagem heurística, grupo de instância $10 \times 5$ . . . . .	45
5.3	Resultados para os sete grupos de instâncias utilizando a heurística. . . . .	47
5.4	Comparação dos resultados para o método híbrido. . . . .	47

# Lista de Abreviaturas e Siglas

ILS	: Iterated Local Search;
AG	: Algoritmo Genético;
SA	: Simulated Annealing;
ALNS	: Adaptive Large Neighborhood Search;
LC	: Lista de Candidatos;
MSILS	: Multi-Start Iterated Local Search;
RVND	: Random Variable Neighborhood Descent;
VND	: Variable Neighborhood Descent;
PLI	: Programação Linear Inteira;
MILP	: Mixed Integer Linear Programming;
B&C	: Branch-and-Cut;
BC&P	: Branch-and-Cut-and-Price;
PCV	: Problema do Caixeiro Viajante;
VRP	: Vehicle Routing Problem;
PRV	: Problema de Roteamento de Veículos;
VRSP	: Vehicle Routing and Scheduling Problem;
SDVRP	: Split Delivery Vehicle Routing Problem;
TDVRP	: Time-Dependent Vehicle Routing Problem;
PRP	: Pollution Routing Problem;
TDPRP	: Time-Dependent Pollution Routing Problem;
GVRP	: Green Vehicle Routing Problem;
GVRSP	: Green Vehicle Routing and Scheduling Problem;
HGVRSP	: Heterogeneous Green Vehicle Routing and Scheduling Problem
TCC	: Taxa de Consumo de Combustível;
CO <sub>2</sub>	: Dióxido de carbono;

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	O Problema de Roteamento de Veículos . . . . .	2
<b>2</b>	<b>O Problema de Roteamento de Veículos Heterogêneos voltado a emissões de <math>CO_2</math></b>	<b>5</b>
2.1	Tipologia do Problema . . . . .	5
2.2	Definição do Problema . . . . .	7
2.3	Exemplo Prático . . . . .	12
<b>3</b>	<b>Revisão da Literatura</b>	<b>16</b>
3.1	Abordagens Exatas . . . . .	16
3.2	Abordagens Heurísticas . . . . .	18
<b>4</b>	<b>Metodologia</b>	<b>20</b>
4.1	Formulação Linear Inteira . . . . .	20
4.2	Heurística Multi-Start Baseada em ILS . . . . .	22
4.2.1	Representação da Solução . . . . .	23
4.2.2	Método Construtivo . . . . .	27
4.2.3	Busca Local . . . . .	29
4.2.3.1	Estruturas de Dados Auxiliares (EDA) . . . . .	30
4.2.3.2	Tratamento das Inviabilidades usando as EDAs . . . . .	32
4.2.3.3	Vizinhanças Inter-rotas . . . . .	35
4.2.3.4	Vizinhanças Intrarotas . . . . .	37
4.2.3.5	Vizinhanças para o <i>Split Delivery</i> . . . . .	38

---

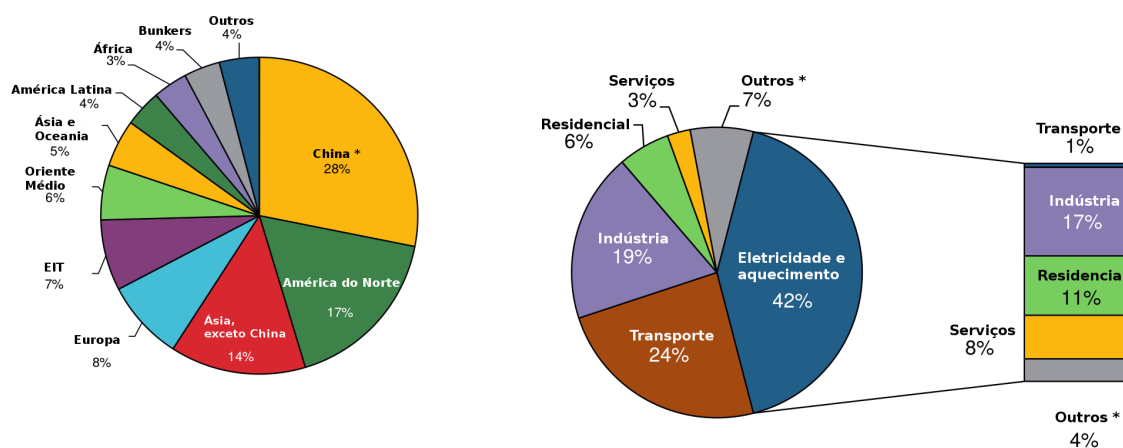
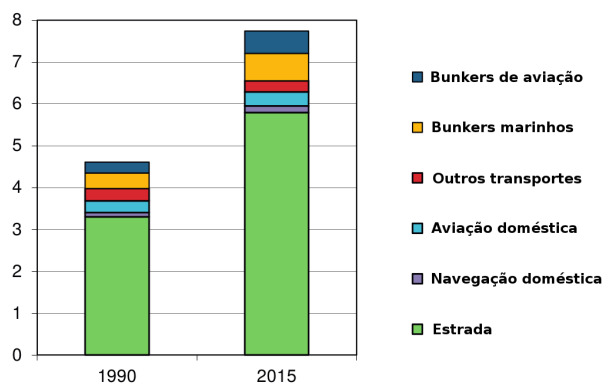
4.2.3.6	Vizinhanças para o HGVRSP . . . . .	39
4.2.4	Mecanismo de Pertubação . . . . .	40
4.3	Formulação <i>Set Covering</i> . . . . .	40
<b>5</b>	<b>Resultados Computacionais</b>	<b>41</b>
5.1	Abordagem Exata . . . . .	42
5.2	Abordagem Heurística . . . . .	45
<b>6</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>48</b>
	<b>Referências</b>	<b>50</b>

# Capítulo 1

## Introdução

Os impactos ambientais decorrentes das atividades de transporte estão se tornando uma questão de interesse global, motivados não só pelo desejo de gerar uma imagem verde, mas também por razões econômicas [50]. Com a crescente preocupação com os impactos ambientais causados pela sociedade, em especial no setor industrial, estima-se que, num futuro próximo, as emissões de gases do efeito estufa passarão a ter um valor monetário [64]. Neste cenário, empresas poderão emitir uma quantidade máxima de gases causadores do efeito estufa e, caso necessitem emitir uma quantidade maior, deverão pagar uma quantidade monetária pelo montante excedente. Outra vantagem em reduzir as emissões de carbono pode ser avaliada no que diz respeito aos gastos com o setor da saúde, uma vez que altos níveis de  $CO_2$  no ambiente podem contribuir consideravelmente com problemas respiratórios, o que traria uma preocupação a mais com gastos na área da saúde [63].

Segundo dados da Agencia Internacional de Energia - [27], a China concentra 28% das emissões mundiais de  $CO_2$ . Na América do Norte, Europa e América Latina esses percentuais são de 17%, 8% e 4%, respectivamente (vide Fig. 1.1(a)). O relatório ainda destaca que as maiores emissões de  $CO_2$  em 2015 concentraram-se nos setores energético e de transporte, que em conjunto são responsáveis por quase 2/3 das emissões no mundo (vide Fig. 1.1(b)). Apesar dos esforços para limitar as emissões no transporte internacional, o setor de transportes teve um aumento de 68% entre 1990 e 2015, sendo que, o setor de transporte rodoviário correspondeu a quase três quartos dessas emissões (vide Fig. 1.1(c)). Ainda segundo [27], na Europa, cerca de 94% das emissões de  $CO_2$  são por conta do setor de transporte rodoviário. Nos Estados Unidos, China e Brasil esses percentuais são de 86%, 78% e 89%, respectivamente.

(a) Emissões de  $CO_2$  por região, 2015(b) Emissões mundiais de  $CO_2$  por setor, 2015GtCO<sub>2</sub>(c) Emissões de  $CO_2$  no setor de transporte, 1990-2015Figura 1.1: Emissões de  $CO_2$  por combustíveis fósseis, Fonte: [27].

Atualmente, há uma mobilização de alguns governos para responsabilizar empresas pelos custos associados aos impactos ambientais, na grande maioria expressa pelo cenário global. Tal situação pode ser observada na Europa, onde verifica-se uma barreira aos produtos importados da China que não tenham certificados que garantam o comprometimento das empresas com o meio ambiente [36]. Neste sentido, deve existir uma preocupação não só com a Logística de Distribuição, mas também com o gerenciamento da emissão de poluentes junto a comunidade científica que estuda problemas de otimização relacionados ao tema.

## 1.1 O Problema de Roteamento de Veículos

O Problema de Roteamento de Veículos, ou "Vehicle Routing Problem" (VRP), foi tratado primeiramente no trabalho "*The Truck Dispatching Problem*" de [15]. O trabalho tratava do problema de distribuição de combustíveis e tinha por objetivo determinar o menor custo

de transporte para atender um conjunto de postos de revenda. Desde a sua introdução por [15], o VRP tem sido extensivamente estudado na literatura devido à sua importância prática e econômica em áreas de transporte, telecomunicações e planejamento de custos.

Mais recentemente uma nova variante do VRP foi introduzida na literatura, o *Time-Dependent Vehicle Routing Problem* (TDVRP) foi formalmente proposto por [9]. No TDVRP, o tempo necessário para um veículo se deslocar ao longo de uma estrada varia de acordo com o período de tempo em que a viagem é realizada. O horizonte de tempo é dividido em períodos, onde cada período representa um intervalo de tempo durante um dia. Por exemplo, o período 1 equivale ao período da manhã (8h às 11h), período 2 define o intervalo próximo das 11h às 13h (comumente chamado de hora do *rush*), período 3 equivale ao período da tarde (13h às 18h) e assim por diante. Para cada período, temos previamente definido uma velocidade para percorrer uma estrada, esta velocidade prefixada é baseada nas médias de velocidade obtidas naquela estrada. Essa variação de velocidades para cada período tenta simular um sistema real de trânsito, desse modo, baixas velocidades tendem a representar congestionamentos. Assim, quanto mais intervalos de tempo com suas respectivas velocidades, mais real é o sistema.

Outra característica interessante vista em problemas de roteamento consiste no agendamento dos horários de acesso aos clientes, dado que os clientes têm requisitos específicos de tempo de serviço. Essa variante do VRP é referenciada na literatura como Problema de Roteamento e Agendas de Veículos (do termo em inglês, *Vehicle Routing and Scheduling Problem* - VRSP) e foi definida pela primeira vez por [6]. Além disso, ao permitirmos que o atendimento do cliente seja realizado por mais de um veículo, temos como consequência direta que os clientes podem ter demandas superiores à capacidade do veículo. Esta outra variante do VRP foi inicialmente referenciada na literatura como *Split Delivery Vehicle Routing Problem* (SDVRP) [17].

Um dos grandes desafios da atualidade, no escopo de *Smart Cities* e *Green Computing*, está relacionado ao planejamento das redes de transporte urbano de forma a minimizar o impacto causado ao meio ambiente. Neste sentido, considerando questões ambientais no VRP, tais como emissão de gases do efeito estufa, especialmente o  $CO_2$ , além de fatores como ruído, poluição dos solos, acidificação, etc., foi definida uma variante do VRP, denominada *Green Vehicle Routing Problem* (GVRP). O GVRP, inicialmente proposto em [5], pode ser definido como uma variante do VRP que visa minimizar as emissões de carbono emitidas pela frota de veículos através da criação de roteamentos e agendamentos eficientes. As rotas definidas no GVRP levam em consideração o horizonte de



tempo ("time-dependent") para melhor simular um cenário real. Além disso, é permitido que o veículo pare na estrada, desse modo, um veículo poderá percorrer um trajeto em múltiplos períodos de tempo. A tarefa de resolver o GVRSP não é trivial, uma vez que sua versão de otimização é  $\mathcal{NP}$ -Difícil [38], i.e. não se conhece algoritmo determinístico capaz de encontrar soluções ótimas em tempo polinomial. Mesmo que os métodos exatos sejam capazes de encontrar soluções ótimas e provar sua otimalidade, a utilização de meta-heurísticas que encontram resultados viáveis de boa qualidade com um baixo custo computacional surge como alternativa para a resolução do GVRSP.

Recentemente, Xiao e Konak [66] apresentaram uma formulação para o GVRSP com restrições de janela de tempo, frota de veículos heterogênea e múltiplas paradas durante o percurso, denominado a nova variante de HGVRSP. Os autores propuseram um método híbrido baseado em "Variable Neighborhood Search"(VND) e Programação Linear Inteira (PLI). Seguindo seus mesmos preceitos, o presente trabalho aborda o problema HGVRSP mas agora considerando um atendimento fracionado, permitindo que o cliente seja atendido por mais de um veículo. Como métodos de resolução do HGVRSP, o presente trabalho propõe primeiro uma abordagem exata que utiliza uma nova formulação matemática que adiciona restrições específicas do HGVRSP assim como restrições específicas para o problema com entregas fracionadas. Além da abordagem exata, é apresentada uma abordagem heurística baseada na meta-heurística *Multi-Start Iterated Local Search* (MS-ILS). Por fim, apresentamos uma abordagem híbrida que utiliza um método exato baseado no Problema de Cobertura de Conjuntos como uma forma de intensificação das soluções encontradas pela meta-heurística, uma extensão do trabalho proposto por [43].

O trabalho está organizado da seguinte forma: o Capítulo 2 apresenta uma breve descrição das principais características do HGVRSP, além de uma definição formal da variante a ser abordada; o Capítulo 3 apresenta uma revisão bibliográfica com a finalidade de observar o atual estado-da-arte para o problema; as metodologias propostas, assim como suas componentes, são introduzidas no Capítulo 4; os resultados computacionais obtidos pelas abordagens são discutidos no Capítulo 5. Por fim, uma breve conclusão e sugestões para trabalhos futuros são realizados no Capítulo 6.

## Capítulo 2

# O Problema de Roteamento de Veículos Heterogêneos voltado a emissões de $CO_2$

Neste capítulo são mostrados os conceitos e definições acerca do HGVRSP. Um breve resumo das principais características do problema é apresentado na Seção 2.1. Na Seção 2.2 é apresentada a definição formal do problema. Por fim, um exemplo prático é descrito na Seção 2.3 com a finalidade de facilitar a compreensão dos diversos aspectos apresentados.

### 2.1 Tipologia do Problema

A tarefa de identificar variantes para o HGVRSP não é considerada simples, uma vez que há diversas características associadas ao problema. A Tabela 2.1 apresenta uma classificação geral de acordo com os elementos mais influentes para o problema.

ATRIBUTOS	HGVRSP	ALTERNATIVAS	
Topologia do Grafo	Não-completo	Completo	-
Horizonte de Planejamento	Finito	Infinito	-
Disponibilidade do Veículo	Parcial	Total	-
Entrega do Veículo	Irrestrita	Restrita	-
Custo de Viagem	Assimétrico	Simétrico	-
Velocidade por arco	Fixa	Variável de decisão	-
Composição da Frota	Heterogênea	Homogênea	Irrestrita
Tamanho da Frota	Fornecido	Variável de decisão	Irrestrito

Tabela 2.1: Elementos para a classificação do HGVRSP.

O primeiro atributo da Tabela 2.1 define a topologia do grafo com relação aos arcos, tal topologia poderá ser completa, quando é possível acessar qualquer vértice a partir de outro vértice; não-completo, quando não há uma conexão direta entre um par qualquer de vértices. O horizonte de planejamento é definido como infinito, quando visa um plano

de distribuição de longo prazo, ou finito, no caso em que uma situação de curto prazo é tratada. Já a disponibilidade do veículo pode ser total, quando qualquer veículo pode ser utilizado a qualquer instante no horizonte de tempo, ou parcial, quando há restrição de tempo para utilização dos veículos. O mesmo se aplica aos locais de entrega utilizando um tipo de veículo, podendo esta entrega ser realizada por qualquer tipo de veículo (irrestrita) ou devendo ser realizada por um conjunto específico de veículos (restrito).

O custo de viagem pode ser classificado em simétrico ou assimétrico, sendo que esse custo também pode ser expresso em função do tempo, distância, latência, consumo de combustível, custos com motoristas ou emissão de gases do efeito estufa, especialmente o  $CO_2$ , além de fatores como ruído, poluição dos solos, acidificação, etc.

A velocidade associada ao arco é outro elemento essencial para definir os custos, sendo esta prefixada para cada arco ou definida como uma variável de decisão. Deste modo, a velocidade prefixada é um dado de entrada, neste trabalho em específico temos um conjunto de velocidades associado a cada arco, o tamanho desse conjunto de velocidades varia de acordo com o número de períodos definido na instância. Por outro lado, há variações em que temos uma velocidade máxima e mínima para percorrer um arco em certo período, assim, a velocidade é considerada como uma variável de decisão.

Em relação ao transporte das mercadorias, a frota de veículos pode ser do tipo homogênea, heterogênea ou mista. Em alguns trabalhos o GVRSP é referenciado como HGVRSP (do termo em inglês, *Heterogeneous Green Vehicle Routing and Scheduling Problem*) pelo fato de usar uma frota heterogênea no roteamento. Os fatores de heterogeneidade entre os veículos podem ser definidos pela capacidade do veículo, taxa de consumo de combustível, disponibilidade de uso em certos horários ou até mesmo a capacidade do tanque de combustível. Ainda sobre a frota, o número de veículos pode ser previamente fornecido, determinado por uma variável de decisão ou ser irrestrito.

Novas variantes que possam surgir de acordo com as características mostradas na Tabela 2.1 podem usar soluções do HGVRSP como limitantes. Como exemplo, a variante menos restrita que permite entregas fracionadas aos clientes, denominada de HGVRSP-Split, poderá ter seu custo de solução menor ou igual a uma solução do HGVRSP para os mesmos dados de entrada, logo a solução ótima do problema HGVRSP pode ser vista como um limitante superior para a solução ótima do HGVRSP-Split. O mesmo é válido para as variantes assumindo a velocidade como variável de decisão, desta forma haveria uma otimização dos custos de carbono assim como as melhores velocidades para percorrer os arcos dentro dos seus respectivos períodos de tempo.

## 2.2 Definição do Problema

O HGVRSP é uma variante do VRP com a adição de restrições de dispersão de  $CO_2$ . Desse modo, o problema abordado nesse trabalho pode ser representado por um grafo  $G = (N, A)$ , onde  $N = \{0, 1, \dots, n\}$  é o conjunto de vértices que representam os  $n$  clientes (incluindo depósito 0) e  $A = \{(i, j) : i, j \in N\}$  o conjunto de arcos, sendo que o arco representa o trajeto entre dois clientes. Seja o conjunto restrito de clientes definido por  $N' = N \setminus \{0\}$ , assim, cada cliente  $i$  possui uma demanda positiva  $R_i$  ( $i \in N' \mid R_0 = 0$ ), além disso, o depósito assume uma demanda nula. Por último, uma distância de viagem  $D_{ij}$  (assimétrico) está associado a cada arco  $(i, j) \in A$ .

Essa variante considera que no máximo  $q$  veículos, partindo de um único depósito, têm uma emissão de  $CO_2$  quando utilizados. Dentre os  $q$  veículos, alguns são utilizados para servir os  $n$  clientes. O cliente  $i$  possui uma janela de tempo  $[S_i, E_i]$  para atendimento e um tempo de serviço  $s_i$  para descarga da sua respectiva demanda. O horizonte de planejamento é particionado em um conjunto de períodos  $K = \{[b_1, e_1], [b_2, e_2], \dots, [b_m, e_m]\}$ , onde  $m$ ,  $b_k$  e  $e_k$  representam respectivamente a quantidade de períodos de tempo, início e fim do período  $k$ . O arco  $(i, j)$  possui condições variáveis de tráfego, i.e., uma velocidade  $v_{ij}^k$  no trajeto  $(i, j) \in A$  é prefixada em cada período  $k \in K$ . Por fim, cada veículo  $v$  possui uma capacidade  $C_v$  e trará consigo a fração de demanda, ou demanda completa, do cliente  $i$ , demanda essa representada por  $y_i^v$ .

A emissão de  $CO_2$  do veículo  $v$  é baseada na Taxa de Consumo de Combustível (TCC) em função da velocidade  $v_{ij}^k$  e peso das cargas  $f_{ij}^v$  do veículo  $v$  no arco  $(i, j)$ . A função TCC, denotada pela variável  $g_{ijk}^v$ , usa coeficientes específicos para cada tipo de veículo para estimar o consumo de combustível. Foram utilizados nesse trabalho dois tipos de veículos com capacidades ( $C_v$ ), coeficientes ( $\alpha_v$ ,  $\beta_v$  e  $\gamma_v$ ), disponibilidade de tempo  $[F_v, T_v]$  e capacidade de tanque de combustível ( $L_v$ ) diferentes. A descrição completa dos veículos bem como outros modelos de emissão podem ser checados em [66]. Por fim, a função TCC pode ser melhor compreendida pela Equação 2.1.

$$g_{ijk}^v = TCC_v(v_{ij}^k, f_{i,j}^v) = \alpha_v \times (v_{ij}^k)^{-1} + \beta_v \times (v_{ij}^k)^2 + \gamma_v + \varphi_v \times f_{ij}^v \quad (2.1)$$

Os parâmetros da Equação 2.1 baseiam-se nos trabalhos desenvolvidos por [16] e [21], esses dois últimos trabalhos basearam-se nos modelos de emissões desenvolvidos por [4] e [3]. Em todos esses trabalhos, cria-se uma estimativa das taxas de consumo de

combustível, viajando numa estrada sem declive, dos veículos em função da velocidade de deslocamento  $v$  e da carga de deslocamento  $f$ . Logo a seguir temos a descrição e notação usada assim como as variáveis de decisão definidas para a modelo do problema HGVRSP.

### Parâmetros:

- $q$ : Número total de veículos
- $v$ : Índice do conjunto de veículos,  
 $v = 0, 1, \dots, q - 1$
- $M$ : Conjunto de veículos heterogêneos
- $n$ : Número total de clientes
- $R_n$ : Conjunto de demandas dos  $n$  clientes
- $i, j$ : Índice dos vértices/clientes, o depósito é representado pelo vértice 0
- $A$ : Conjunto de arcos formado por todos os pares de vértices,  $(i, j) \in A$ , sendo  $D_{ij}$  a distância do arco  $(i, j)$
- $L_v$ : Capacidade do tanque de combustível do veículo  $v$
- $[F_v, T_v]$ : Início e fim da janela de tempo do veículo  $v$
- $Q_v$ : Capacidade do veículo  $v$
- $m$ : Número total de períodos
- $k$ : Índice do período de tempo,  
 $k = 0, 1, \dots, m-1$
- $K$ : Conjunto de períodos,  $k \in K$
- $[b_k, e_k]$ : Tempo de início e término do período  $k$ ;  $k = 0, 1, \dots, m-1$
- $[S_i, E_i]$ : Início e fim da janela de tempo do cliente  $i$
- $s_i$ : Tempo de serviço do cliente  $i$
- $v_{ijk}$ : Velocidade no arco  $(i, j)$  no período de tempo  $k$  (pré-calculado)
- $g_{ijk}^v$ : Taxa de consumo de combustível do veículo  $v$  no arco  $(i, j)$  e período de tempo  $k$ , valor pré-calculado para um veículo vazio.
- $\mathcal{M}$ : Número suficientemente grande

### Variáveis de decisão:

- $x_{ijk}^v = \begin{cases} 1, & \text{se o veículo } v \text{ percorre o} \\ & \text{arco } (i, j) \text{ no período } k, \text{ não} \\ & \text{necessariamente todo o arco} \\ 0, & \text{c.c.} \end{cases}$
- $\widehat{X}_{ij} = \begin{cases} 1, & \text{se o arco } (i, j) \text{ é percorrido} \\ & \text{por algum veículo} \\ 0, & \text{c.c.} \end{cases}$
- $d_{ijk}^v = \begin{cases} \text{Distância percorrida pelo veículo} \\ v \text{ no arco } (i, j) \text{ no período } k \end{cases}$
- $\bar{X}_{ij}^v = \begin{cases} 1, & \text{se o veículo } v \text{ percorre} \\ & \text{o arco } (i, j) \text{ em algum tempo} \\ 0, & \text{c.c.} \end{cases}$
- $t_{ijk}^v = \begin{cases} \text{Tempo total em que o veículo } v \text{ se} \\ \text{desloca no arco } (i, j) \text{ e no período } k \end{cases}$
- $l_i/a_i = \begin{cases} \text{Tempo de } \mathbf{partida} \text{ e } \mathbf{chegada} \\ \text{no vértice } i \end{cases}$
- $f_{ij}^v = \begin{cases} \text{Carga do veículo } v \text{ no arco } (i, j) \end{cases}$

No HGVRSP, uma velocidade  $v_{ij}^k$  é fixada para cada arco  $(i, j)$  e período de tempo  $k$ , tais dados estão disponíveis nas instâncias reportadas em [66]. O artigo também fornece dados específicos do veículo (por exemplo, eficiência do motor, peso e capacidade) e as

condições do curso do arco (por exemplo, conjunto de  $k$  velocidades e distância). As emissões de  $CO_2$  são calculadas para cada arco em cada período de tempo, tais emissões podem ser calculadas utilizando outros modelos de emissão da literatura, como os propostos em [4], [3] ou [16]. Para cada arco  $(i, j)$ , a velocidade de deslocamento em um período é constante, representando a velocidade média durante o período. Este pressuposto é bem justificado no caso de estradas urbanas e suburbanas, onde a velocidade média de deslocamento depende da hora e do dia, o que de certa forma simula as condições da estrada em certas ocasiões. Para o melhor entendimento das restrições do HGVRSP, o modelo reportado em [66] seguido de uma descrição detalhada está exposta a seguir:

**(MILP) Minimizar**

$$\text{Min } CO_2 = \sum_{(i,j) \in A} \sum_{k \in K} \sum_{v \in M} c_v g_{ijk}^v d_{ijk}^v + \sum_{(i,j) \in A} \sum_{v \in M} c_v \varphi_v f_{ij}^v D_{ij} \quad (2.2)$$

**Sujeito a:**

$$d_{ijk}^v \leq D_{ij} \times x_{ijk}^v, \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (2.3)$$

$$x_{ijk}^v \leq \bar{X}_{ij}^v, \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (2.4)$$

$$\bar{X}_{ij}^v \leq \sum_{k \in K} x_{ijk}^v, \quad \forall (i, j) \in A; \quad \forall v \in M \quad (2.5)$$

$$\widehat{X}_{ij} = \sum_{v \in M} \bar{X}_{ij}^v, \quad \forall (i, j) \in A \quad (2.6)$$

$$\sum_{j \in N'} \bar{X}_{0j}^v \leq 1, \quad \forall v \in M \quad (2.7)$$

$$\sum_{j \in N} \widehat{X}_{ij} = 1, \quad \forall i \in N' \quad (2.8)$$

$$\sum_{i \in N} \widehat{X}_{ij} = 1, \quad \forall j \in N' \quad (2.9)$$

$$\sum_{i \in N; (i,p) \in A} \bar{X}_{ip}^v - \sum_{p \in N; (p,j) \in A} \bar{X}_{pj}^v = 0, \quad \forall p \in N'; \quad \forall v \in M \quad (2.10)$$

$$\widehat{X}_{ij} \times D_{ij} = \sum_{k \in K} \sum_{v \in M} d_{ijk}^v, \quad \forall (i, j) \in A \quad (2.11)$$

$$x_{ijk}^v \leq \mathcal{M} \times d_{ijk}^v, \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (2.12)$$

$$x_{jpk_1}^v \leq 2 - x_{ijk_2}^v - \bar{X}_{ij}^v, \quad \forall (i, j), (j, p) \in A; \forall v \in M; \forall (k_1, k_2) \in K | k_1 < k_2 \quad (2.13)$$

$$t_{ijk}^v = \frac{d_{ijk}^v}{v_{ijk}} \times 60, \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (2.14)$$

$$\sum_{(i,j) \in A} t_{ijk}^v \leq (e_k - b_k), \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.15)$$

$$l_i \leq e_k - t_{ijk}^v + \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall \quad (i,j) \in A; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.16)$$

$$a_j \geq b_k + t_{ijk}^v - \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall \quad (i,j) \in A; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.17)$$

$$a_j \geq l_i + \sum_{k \in K} \sum_{v \in M} t_{ijk}^v - \mathbf{e}_m \times (1 - \widehat{X}_{ij}), \quad \forall \quad (i,j) \in A \quad (2.18)$$

$$a_i + s_i \leq l_i, \quad \forall \quad i \in N' \quad (2.19)$$

$$S_i \leq a_i \leq E_i, \quad \forall \quad i \in N' \quad (2.20)$$

$$\sum_{i \in N; i \neq j} \sum_{v \in M} f_{ij}^v - \sum_{i \in N; i \neq j} \sum_{v \in M} f_{ji}^v = R_j, \quad \forall \quad j \in N' \quad (2.21)$$

$$f_{ij}^v \leq C_v \times \bar{X}_{ij}^v, \quad \forall \quad (i,j) \in A; \quad \forall \quad v \in M \quad (2.22)$$

$$L_v \geq \sum_{(i,j) \in A} \sum_{k \in K} g_{ijk}^v \times d_{ijk}^v + \sum_{(i,j) \in A} \varphi_v \times f_{ij}^v \times D_{ij}, \quad \forall \quad v \in M \quad (2.23)$$

$$F_v \leq e_k - t_{0jk}^v + e_m \times (1 - x_{0jk}^v) \quad \forall \quad j \in N'; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.24)$$

$$F_v \leq a_j - t_{0jk}^v + e_m \times (1 - x_{0jk}^v) \quad \forall \quad j \in N'; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.25)$$

$$T_v \geq b_k + t_{i0k}^v - e_m \times (1 - x_{i0k}^v) \quad \forall \quad i \in N'; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.26)$$

$$T_v \geq l_i + t_{i0k}^v - e_m \times (1 - x_{i0k}^v) \quad \forall \quad i \in N'; \quad \forall \quad k \in K; \quad \forall \quad v \in M \quad (2.27)$$

### Domínio das variáveis de decisão:

$$x_{ijk}^v \in \{0, 1\}, \quad \forall \quad (i,j) \in A; \quad \forall \quad v \in M; \quad \forall \quad k \in K \quad (2.28)$$

$$\bar{X}_{ij}^v \in \{0, 1\}, \quad \forall \quad (i,j) \in A; \quad \forall \quad v \in M \quad (2.29)$$

$$\widehat{X}_{ij} \in \{0, 1\}, \quad \forall \quad (i,j) \in A \quad (2.30)$$

$$d_{ijk}^v \geq 0, \quad \forall \quad (i,j) \in A; \quad \forall \quad v \in M; \quad \forall \quad k \in K \quad (2.31)$$

$$t_{ijk}^v \geq 0, \quad \forall \quad (i,j) \in A; \quad \forall \quad v \in M; \quad \forall \quad k \in K \quad (2.32)$$

$$l_i \geq 0, \quad \forall \quad i \in N \quad (2.33)$$

$$a_i \geq 0, \quad \forall \quad i \in N \quad (2.34)$$

O lado esquerdo da função objetivo (2.2) minimiza os custos de emissão de carbono (medido em quilogramas -  $kg$ ) em função da distância (medido em quilômetros -  $km$ ) para um veículo vazio. O custo é dado pela taxa de consumo de combustível  $g_{ijk}^v$  multiplicado por  $c_v$ <sup>1</sup>, o que resulta em uma emissão de  $CO_2$  ( $kg$ ), essa emissão é finalmente multiplicada pelo segmento de distância  $d_{ijk}^v$  ( $km$ ), obtendo assim, um custo de  $CO_2$  ( $kg/km$ ). Já o lado direito da função objetivo é voltado para calcular a emissão extra causada pelo peso das cargas do veículo durante o trajeto, novamente temos o resultado da multiplicação de  $c_v$  por  $f_{ij}^v$ ,  $D_{ij}$  e  $\varphi_v$ , sendo esta última variável uma taxa adicional de consumo do veículo  $v$  quando percorre o arco  $(i, j)$  com peso  $f_{ij}^v$ .

A restrição (2.7) garante que o veículo sairá no máximo uma vez do depósito. As restrições (2.10) e (2.13) estabelecem, respectivamente, a conservação de fluxo e eliminação de subciclo. As restrições (2.4), (2.5) e (2.6) asseguram a consistência entre as variáveis  $\widehat{X}_{ij}$ ,  $\bar{X}_{ij}^v$  e  $x_{ijk}^v$ . As restrições (2.3) e (2.11) asseguram a consistência entre as variáveis  $x_{ijk}^v$  e  $d_{ijk}^v$ . A restrição (2.3) garante que, se o veículo  $v$  percorre parte do arco  $(i, j)$  no período  $k$ , então essa fração de distância não será maior que a distância  $D_{ij}$ . Já a restrição (2.11) assegura que, se um arco  $(i, j)$  é percorrido, então a soma das frações de distância  $d_{ijk}^v$  deverá ser igual a distância  $D_{ij}$ . As restrições (2.8) e (2.9) garantem que todo cliente será visitado uma única vez. Por fim, a restrição (2.12) assegura que se um arco  $(i, j)$  for percorrido, então uma distância mínima  $d$  ( $\frac{1}{M}$ ) deverá ser percorrida.

As próximas restrições tratam das variáveis de decisão vinculadas ao tempo. Desse modo, a restrição (2.14) converte a variável  $t_{ijk}^v$  em minutos, já a restrição (2.15) garante que o montante de tempos  $t_{ijk}^v$  percorrido no período  $k$  nunca será maior que o próprio período de tempo  $k$ . A consistência das variáveis  $l_i$  e  $a_j$  é assegurada respectivamente pelas restrições (2.16) e (2.17). A restrição (2.16) garante que se o arco  $(i, j)$  for percorrido no período  $k$  pelo veículo  $v$ , então o tempo de partida do vértice  $i$  deve ser antes do término do período  $k$  menos o tempo de viagem do arco  $(i, j)$  nesse período. Por outro lado, restrição (2.17) garante que, o tempo de chegada do cliente  $j$  deve ser maior ou igual ao início do período  $k$  mais o tempo de viagem do arco  $(i, j)$  nesse período. A restrição (2.18) garante que, para o arco  $(i, j)$ , o tempo de chegada no vértice  $j$  é maior que o tempo de saída do vértice  $i$  mais o tempo de trajeto do arco. A restrição (2.19) assegura que o tempo de saída no vértice  $i$  é maior que o tempo de serviço  $s_i$  mais o tempo de chegada no mesmo vértice  $i$ . Por fim, a restrição (2.20) garante que o tempo de chegada no vértice  $i$  está dentro da janela de tempo.

<sup>1</sup> $c_v$ : Taxa de conversão de consumo de combustível para  $CO_2$  do veículo  $v$ .



A restrição (2.23) trata da capacidade do tanque de combustível do veículo, no lado direito da desigualdade temos duas expressões, a expressão mais a esquerda define o consumo de combustível baseado na distância total percorrida, já expressão a direita define o consumo baseado nas cargas do veículo  $v$ . A restrição (2.21) garante que a diferença entre o fluxo de cargas entrando e saindo no cliente  $i$  seja igual a carga do cliente  $i$ . A restrição (2.22) estabelece que o montante de cargas no arco  $(i, j)$  não excederá a capacidade do veículo. Os pares de restrições (2.24 , 2.25) e (2.26, 2.27) estabelecem, respectivamente, os tempos mínimos de partida ( $F_v$ ) e máximos de retorno ( $T_v$ ) do veículo  $v$ .

## 2.3 Exemplo Prático

Com o propósito de descrever de forma detalhada o HGVRSP, exemplos práticos são apresentados nesta seção. Temos na Figura 2.1 um arco  $(i, j)$  representando o percurso entre o cliente  $i$  e  $j$ , esse percurso está esboçado na forma de *time-line*<sup>2</sup>, onde a linha horizontal que une o cliente  $i$  a  $j$  representa o tempo de percurso, a linha tracejada na vertical mais a esquerda e na direita representam, respectivamente, o início e término do período  $k$ . Cinco intervalos de tempos são definidos, cada intervalo de tempo representa um período, sendo o período  $P0$  (rotulado em verde) definido entre 0 e 108 minutos, período  $P1$  (rotulado em amarelo) definido entre 108 e 216 minutos e assim por diante.

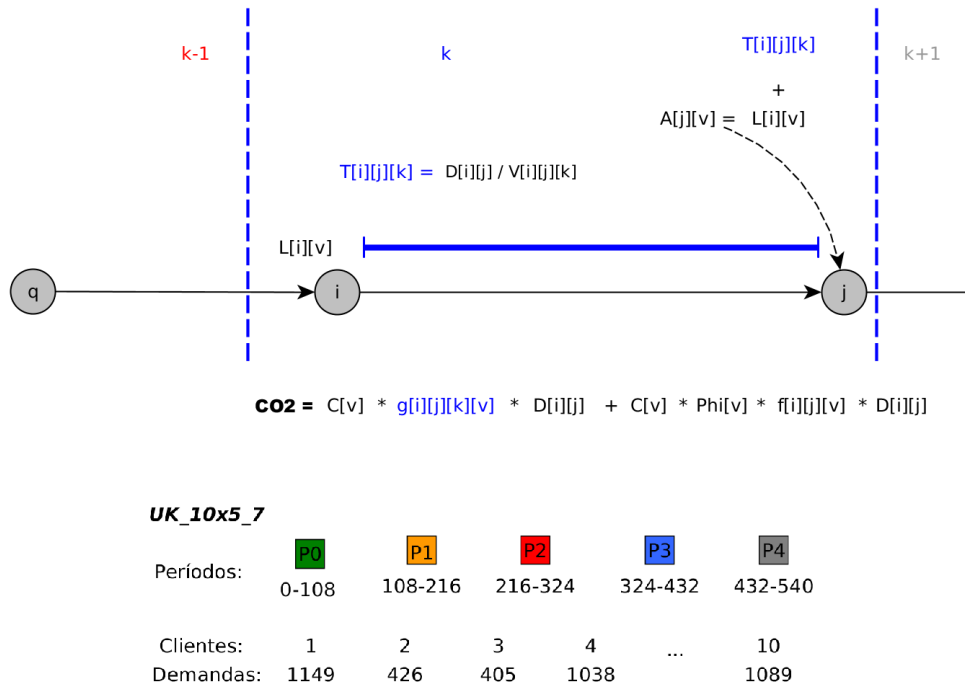


Figura 2.1: Representação de parte da rota  $v$  na forma de *time-line*.

<sup>2</sup>*time-line*: forma gráfica e linear de representar uma sequência de eventos em ordem cronológica.

Ainda avaliando o exemplo da Figura 2.1, um veículo  $v$ , partindo do cliente  $i$  no instante de tempo  $l_i^v$  ( $L[i][v]$ ), percorre todo arco  $(i, j)$  dentro do período de tempo  $k$  (delimitado pelas linhas tracejadas em azul na vertical). Logo, o cálculo de  $CO_2$  (em negrito) do arco  $(i, j)$  é realizado com base na taxa de consumo de combustível  $g_{ijk}^v$  do período  $k$  e peso das cargas  $f_{ij}^v$  presente no veículo  $v$ . Neste sentido, o cálculo do tempo de chegada no cliente  $j$  é o resultado da soma do instante de tempo de saída do cliente  $i$  com o tempo de percurso  $T[i][j][k]$  do arco  $(i, j)$ . O cálculo dos tempos de chegada e saída para os demais clientes da rota  $v$  são realizados de forma análoga, assim, avaliam-se os tempos de saída do cliente anteriormente atendido pela rota  $v$  e assim sucessivamente.

A Figura 2.2 mostra o exemplo de um arco  $(i, j)$  percorrido nos períodos  $k$  e  $k+1$ . Ao contrário da figura anterior, tanto os custos de carbono quanto o tempo de percurso no arco  $(i, j)$  dependem das frações de distâncias percorridas em cada período de tempo. Neste sentido, temos a primeira parte do arco sendo percorrido no período  $k$  (delimitado entre as linhas verticais tracejadas em vermelho e azul) e período  $k+1$  (a partir da linha tracejada em azul na vertical), para cada período calcula-se as frações de tempos ( $T[i][j][k]$  e  $T[i][j][k+1]$ ) e distâncias ( $d[i][j][k]$  e  $d[i][j][k+1]$ ) em que o veículo esteve em movimento afim de avaliar os custos de  $CO_2$  no arco  $(i, j)$  e tempo de chegada no cliente  $j$ .

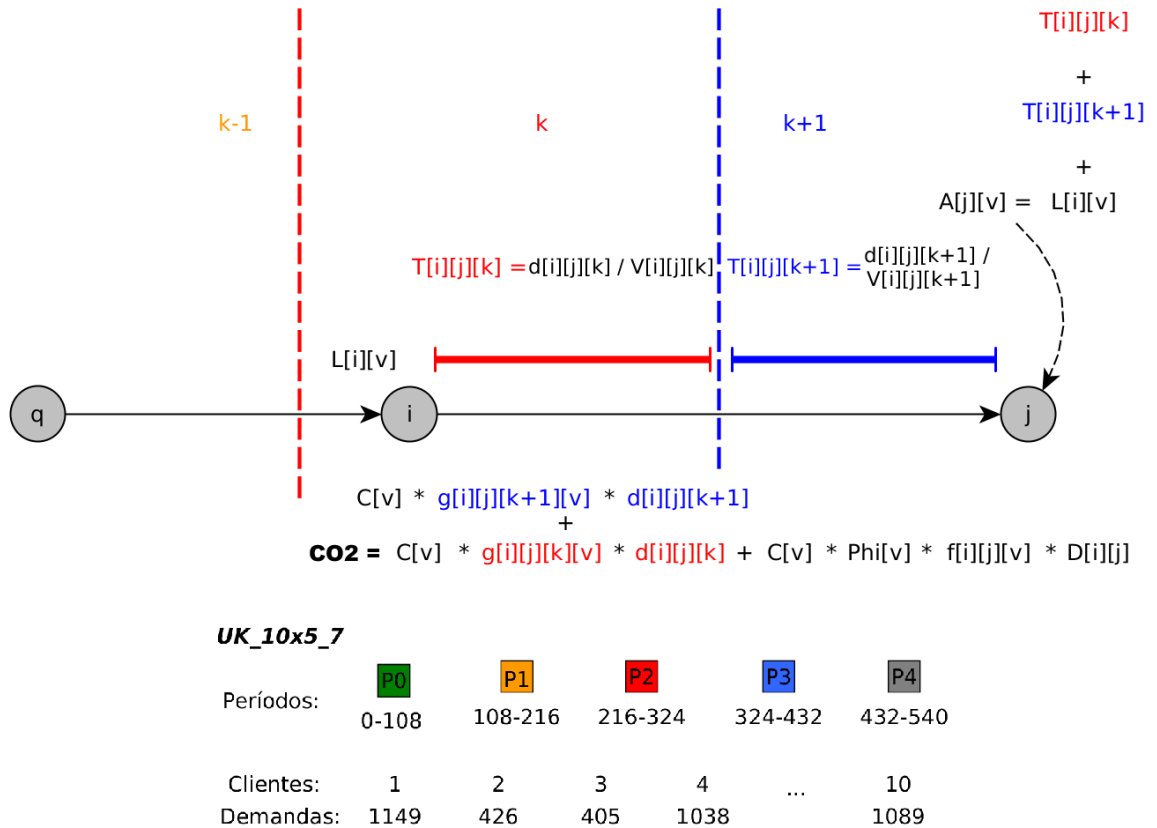


Figura 2.2: Solução com troca de períodos consecutivos  $k$  e  $k+1$ .

A Figura 2.2 ainda mostra o cálculo de  $CO_2$  (em negrito) no arco  $(i, j)$  nos períodos  $k$  (em vermelho) e  $k+1$  (em azul). Em contrapartida, o lado direito do cálculo adiciona um custo extra causado pelas cargas presentes no veículo  $v$  ao percorrer o arco  $(i, j)$ , sendo este último custo dependente apenas do montante de cargas  $f[i][j][v]$  e distância  $D[i][j]$ .

A Figura 2.3 mostra a configuração de parte da rota  $v$  sendo percorrida no período  $k$  e  $k+2$ . Temos no arco  $(i, j)$  o tempo de saída  $l_i^v$  ( $L[i][v]$ ) do cliente  $i$ , tempo de percurso ( $T[i][j][v]$ ) no arco  $(i, j)$  e tempo de saída  $l_j^v$  ( $L[j][v]$ ) do cliente  $j$ . Temos ainda neste cenário três intervalos de períodos: período  $k$  (entre as linhas verticais tracejadas em amarelo e vermelho), período  $k+1$  (entre as linhas verticais tracejadas em vermelho e azul) e período  $k+2$  (a partir da linha tracejada vertical em azul).

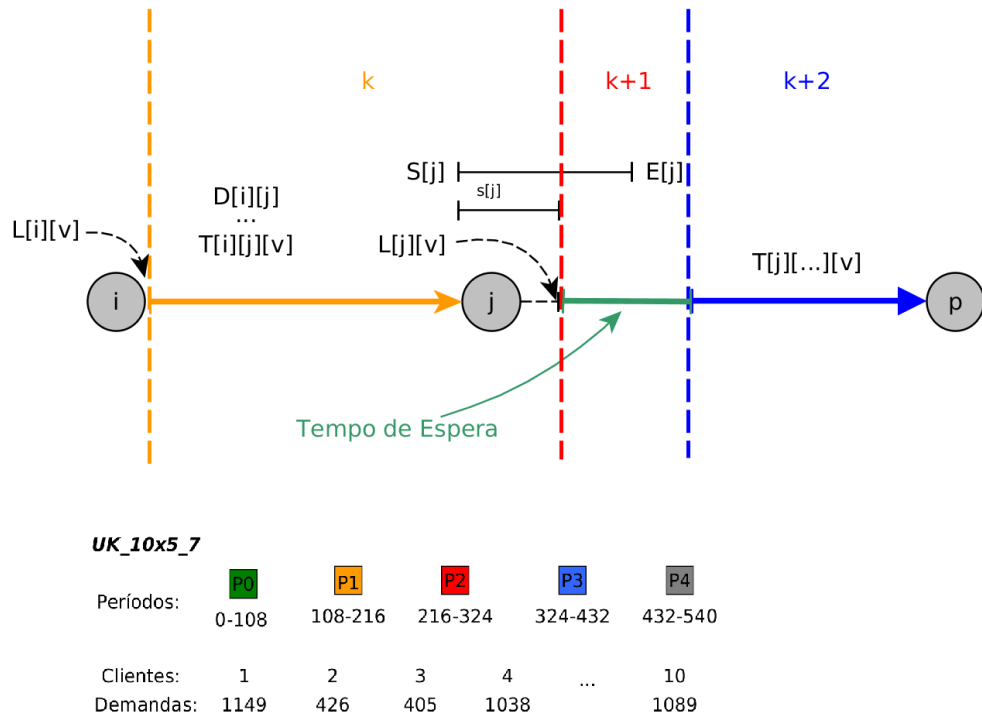


Figura 2.3: Solução com parada entre os períodos  $k$  e  $k+2$ .

Ainda sobre a Figura 2.3, temos o tempo de saída do cliente  $j$  coincidindo com o término do período  $P1$  e início do período  $P2$ . Após o atendimento do cliente  $j$  temos um tempo de parada grande o suficiente para evitar percorrer o arco  $(j, p)$  no período  $k+1$ . Neste sentido, o arco  $(j, p)$  é percorrido logo após o término do período  $k+1$ , ou seja, o período  $k+1$  foi evitado ao longo da rota do veículo  $v$ . Este pressuposto de paradas é bem justificado no caso de estradas em que a velocidade em um dado período é baixa, sendo essa baixa velocidade representando um possível congestionamento. Assim, é ideal evitar esses períodos congestionados aplicando pequenas paradas ao longo do trajeto com o intuito de gerar a menor quantidade possível de  $CO_2$ .

A Figura 2.4 representa o funcionamento do HGVRSP com entrega fracionada em todas as rotas. O depósito, representado pelo vértice 0, armazena três veículos e os clientes estão representados por vértices numerados de 1 a 15. Além disso, na solução apresentada, toda a frota de 3 veículos foi usada utilizando todos os cinco períodos.

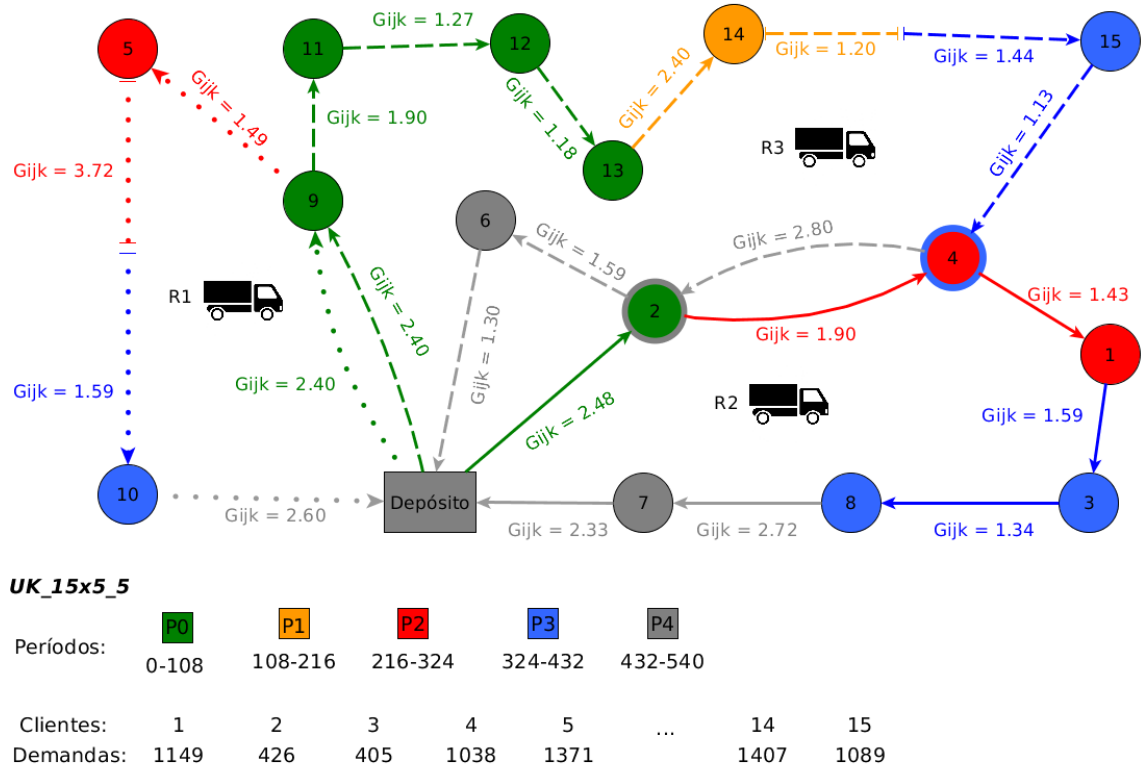


Figura 2.4: Solução para uma instância com 15 clientes, 5 períodos, 3 veículos e emissão de 792.36 kg/km de  $CO_2$ .

No exemplo da Figura 2.4 o arco  $(14, 15)$  é percorrido pelo veículo  $R_3$  em dois períodos. O veículo  $R_3$  percorre parte do arco no período  $P_1$ , para, e espera iniciar o período posterior, neste caso, o  $P_3$  para continuar o trajeto. Uma situação similar ocorre no arco  $(5, 10)$ , entretanto, o veículo  $R_1$  não realiza uma parada no arco. Situações de transição de períodos são comuns em arcos  $(i, j)$  de grandes distâncias ou quando o veículo está percorrendo um arco próximo da transição de um período  $k$  para  $k + 1$ . Vale ressaltar que no arco  $(0, 9)$  temos a mesma taxa de consumo de combustível pois veículos com as mesmas características foram utilizados no período  $P_0$ , o que não necessariamente define o mesmo custo de carbono pois o montante de cargas em cada veículo pode variar. Ainda sobre o arco  $(0, 9)$ , temos neste arco a maior emissão de carbono causado pelas cargas tanto para rota  $R_1$  quanto para rota  $R_3$ . Este fato ocorre por que no primeiro arco de cada rota temos todas as cargas presentes no veículo, situação contrária ocorre no arco  $(10, 0)$ , onde este último arco possui custo zero de carbono causado pelas cargas.

# Capítulo 3

## Revisão da Literatura

Neste capítulo são apresentados os principais trabalhos da literatura que abordam o HG-VRSP. Atualmente há poucos trabalhos que se dedicam à variante tratada, o processo de levantamento bibliográfico teve como objetivo proporcionar uma visão geral sobre o estado-da-arte, ao invés de se concentrar apenas em uma variante específica do HGVRSP. Na Seção 3.1, são listados os diversos trabalhos que propõem abordagens exatas, seguido da Seção 3.2, que discute os trabalhos que fazem uso de heurísticas ou abordagens híbridas. Além das abordagens, são listados alguns trabalhos que reforçam o tema de “Green Computing & Smart Cities” em aplicações reais com algum impacto na sociedade.

### 3.1 Abordagens Exatas

O VRP pode ser definido como um conjunto de problemas de otimização combinatória, tais problemas tem recebido muita atenção nas últimas décadas devido à sua importância prática e econômica em áreas de transporte, telecomunicações e planejamento de custos. Desde a sua introdução por [15], o VRP tem sido extensivamente estudado na literatura, assim, diversas variações do VRP foram criadas para diferentes aplicações, tais como coleta e entrega, entrega dividida e dependente do tempo. Alguns *Surveys* das mais diversas formulações do VRP podem ser encontrados em [37], [59], [23] e [18].

O VRSP foi definido pela primeira vez por [6], no VRSP os clientes têm a adição de requisitos específicos de tempo de serviço. Portanto, uma solução para o VRSP inclui tanto as rotas como os cronogramas dos veículos, onde esses cronogramas indicam os horários de chegada e partida nos clientes. Em uma abordagem exata, [53] apresentou pela primeira vez uma formulação de Programação Inteira Mista para o VRSP com restrições de janela de tempo, onde os clientes impõem restrições de tempo em suas coletas e entregas.

Novas classes de desigualdades válidas a partir de uma formulação foram estendidas para o SDVRP em [44]. Os autores propuseram um algoritmo BC&P, o algoritmo foi aplicado a instâncias com até 288 clientes. Uma abordagem similar foi proposta em [1], os autores também desenvolveram um algoritmo BC&P. Os autores empregaram um procedimento heurístico para obter soluções inteiras a partir das colunas geradas em ambas versões do problema, SDVRP com frota limitada e SDVRP com frota ilimitada.

Recentemente, em [33] os autores propuseram uma abordagem combinando uma meta-heurística com um modelo MILP para o *Pollution Routing Problem* (PRP) e relataram um melhor desempenho do que o algoritmo proposto por [42] para as mesmas instâncias. Em [56], os autores propuseram um modelo MILP para o TDVRP capacitado com o objetivo de minimizar o consumo de combustível. Além do consumo de combustível, os autores também comparam os custos em função da distância e tempo.

Uma formulação matemática para o chamado *Energy Minimizing Vehicle Routing Problem* (EMVRP) foi proposta em [30], o EMVRP tem como objetivo minimizar a soma do produto entre carga e distância pra cada arco das rotas. Abordagens similares, i.e, aquelas que fazem uso da carga do veículo para minimizar o consumo de combustível e/ou emissões de  $CO_2$ , foram apresentadas em [47], [51], [60] e [67].

Os trabalhos mais recentes sobre a otimização das emissões de  $CO_2$  no contexto de GVRP em condições de tráfego com variações de tempo são os seguintes trabalhos: [20, 64, 65, 66]. Nessa perspectiva, um modelo MILP foi proposto para o *Time-Dependent PRP* (TDPRP) com objetivos hierárquicos e uma penalidade de atraso ponderada sob um sistema de emissão de carbono [65]. Os autores avaliaram as emissões totais de  $CO_2$ , tempo de chegada e o tempo total de viagem. Minimizar as emissões de  $CO_2$  é considerado como o objetivo primário no GVRP, porém, há outros objetivos secundários, tais como o tempo total de chegada, que reflete o nível de satisfação do cliente, o tempo total do percurso, que reflete o custo de utilização do condutor e do veículo, e por fim, a distância total, o que reflete diretamente no consumo de combustível.

Com relação ao HGVRSP, as abordagens exatas existentes ainda são muito limitadas, ficando restritas a instâncias de tamanho reduzido. Assim, algoritmos heurísticos parecem ainda ser os mais adequados para tratar instâncias do HGVRSP de tamanhos médios e grandes. Neste sentido, vários autores desenvolveram abordagens baseadas em heurísticas e meta-heurísticas. Os principais métodos encontrados na literatura são apresentados na seção seguinte.

## 3.2 Abordagens Heurísticas

Um algoritmo híbrido para o SDVRP que combina um procedimento construtivo baseado na heurística de *Clarke & Wright* [11] com uma formulação PIM foi desenvolvido em [10]. Sob outra perspectiva, os autores em [8] propuseram uma heurística populacional baseada na meta-heurística *Scatter Search* (SS) [22] para o problema. Nesta heurística a população inicial é gerada por dois procedimentos diferentes.

O algoritmo proposto em [7] para tratar o SDVRP também mantém uma população de soluções durante sua execução, o algoritmo proposto é baseado em Algoritmo Genéticos (AG) [25] e incorpora um procedimento de busca local. Os autores também propuseram um procedimento de gerência de população para controlar a diversidade das soluções. Na fase de busca local, além das estruturas de vizinhanças clássicas do VRP baseadas em trocas e realocações de clientes, também adicionaram três novas estruturas específicas para o SDVRP, sendo uma delas a generalização do *k-Split*.

Nos últimos anos, uma outra abordagem relacionada ao TDVRP foi proposta com o objetivo de minimizar as emissões de  $CO_2$  em sistemas logísticos. Sob esta perspectiva, os autores em [5] apresentaram um PRP em que a velocidade de deslocamento ótima de cada arco  $(i, j)$  é determinada para minimizar uma função objetivo. O problema apresenta restrições de consumo de combustível e emissões de  $CO_2$ . Neste mesmo trabalho, foi proposto um modelo para minimizar o consumo de combustível onde os autores usaram uma abordagem heurística baseada em *Simulated Annealing* para resolver o problema. Neste mesmo contexto, os autores de [21] estenderam o PRP para o TDPRP. O trabalho considerava um tráfego com congestionamento em dois períodos com velocidades prefixadas. O TDPRP também foi abordado em [64], onde os autores propuseram uma abordagem híbrida de algoritmos genéticos e programação dinâmica para resolvê-lo.

A maioria das versões *time-dependent* assumem que as velocidades no arco  $(i, j)$  são fixas e iguais às velocidades do instante de partida de  $i$ , de modo que, se a velocidade no instante de tempo  $k + 1$  for maior que a velocidade no instante de tempo  $k$ , uma viagem iniciando mais tarde, no período  $k + 1$ , possivelmente terminaria antes que uma viagem iniciando mais cedo, no período  $k$ , ocorrendo uma situação de “passing” [34]. Diante do exposto, [34] propôs um modelo para o TDVRP tratando a duração da viagem e a velocidade dos veículos como dependentes do instante de tempo em que a viagem é realizada e os custos associados ao consumo de combustível em função da quantidade de carga transportada pelo veículo. Para solucionar o problema o autor utilizou a meta-

heurística *Simulated Annealing* [32], obtendo uma redução de 24.61% no consumo de combustível quando comparado com as soluções obtidas considerando apenas a redução do tempo total de viagem e de 22.69% em relação ao modelo minimizando a distância.

No ano seguinte, o mesmo problema foi resolvido através de um algoritmo de Busca Tabu por [35]. Mais tarde, [29] também abordaram o TDVRP considerando propriedades de *non-passing*, porém tratando os custos como função da velocidade, da carga e da duração das rotas, denominando-o de *Emissions-based Time-Dependent Vehicle Routing Problem* (E-TDVRP). As velocidades são escolhidas de acordo com um nível ótimo de emissões, exceto em horários de congestionamento onde são assumidas como constantes.

Em [46] os autores resolveram instâncias similares às tratadas em [19] através de dois algoritmos: uma heurística baseada em busca local multi-start para geração das rotas e um algoritmo recursivo suavizado (ARS) para otimizar a velocidade. O ARS mostrou-se mais eficiente que o apresentado em [19] quando a função de consumo de combustível depende apenas da velocidade, porém, ao considerar outras variáveis, a abordagem não mostrou-se viável. A viabilidade do ARS só foi provada posteriormente em [26], onde os autores mostram que as velocidades ótimas podem ser encontradas em tempo quadrático.

A seguir, em [16] os autores abordaram o PRP com uma heurística em dois estágios. Os autores inicialmente resolveram o VRSP através da meta-heurística ALNS utilizando 5 operadores de inserção e 12 de remoção, dos quais 4 são novos (1 de inserção e 3 de remoção), tendo a meta-heurística *Simulated Annealing* como busca local. No segundo estágio, os autores utilizaram um algoritmo exato recursivo para otimizar a velocidade em cada arco de uma dada solução, de modo a minimizar o consumo de combustível e os custos com motoristas. Para verificar a eficiência do algoritmo ALNS, foram resolvidas as instâncias propostas no trabalho de [54].

No contexto de temas afins referentes a “Green Computing & Smart Cities”, alguns autores desse trabalho possuem experiência comprovada em publicações nesta área [12, 13, 43]. Além desses trabalhos, pode-se citar o *Survey* proposto por [39] como introdução ao GVRP e suas relações com outras versões do TDVRP. Também vale citar o trabalho proposto por [14] como introdução ao tema de *Green Cities*.



# Capítulo 4

## Metodologia

Neste capítulo serão apresentadas as abordagens utilizadas para resolver o HGVRSP. Duas abordagens serão descritas, uma abordagem exata e um método híbrido que utiliza uma meta-heurística e um método exato baseado no Problema de Cobertura de Conjuntos. A abordagem exata faz uso de uma nova formulação de Programação Linear Inteira (PLI) onde foram adaptadas certas restrições afim de contemplar o HGVRSP na sua variante com *Split*. Por outro lado, a meta-heurística proposta é baseada no *Iterated Local Search* (ILS) com comportamento *Multi-Start*, além de incorporar na etapa de busca local o mecanismo *Random Variable Neighborhood Descent* (RVND). A Seção 4.1 apresenta o modelo MILP, já a meta-heurística híbrida MS-ILS tem seus componentes heurísticos detalhados na Seção 4.2 e, por fim, o componente exato é exposto na Seção 4.3.

### 4.1 Formulação Linear Inteira

Logo a seguir, é feita uma breve descrição do modelo a fim de detalhar melhor as restrições do problema. Nesse trabalho, o modelo MILP foi baseado no modelo proposto em [66] (apresentado na Seção 2.2) e adicionado restrições específicas do SDVRP descritas no *Survey* apresentado em [2], os dados de entrada e variáveis de decisão estão descritos na Seção 2.2 do Capítulo 2.

Para contemplar a variante do SDVRP, além de alterações no modelo reportado em [66], variáveis foram adicionadas, retiradas e outras precisaram de um redimensionamento. A variável  $y_i^v$  foi adicionada com o objetivo de armazenar as frações de demandas ao longo de uma rota, i.e., a quantidade de demanda que o veículo  $v$  atende no cliente  $i$ . Além disso, a variável  $\widehat{X}_{ij}$  foi retirada já que mais de um veículo pode atender o mesmo cliente agora. Outra alteração necessária está vinculada as variáveis  $l_i$  e  $a_i$  que agora são representadas

respectivamente por  $l_i^v$  e  $a_i^v$ , tais variáveis agora armazenam os tempos de saída e chegada do veículo  $v$  no cliente  $i$ . As alterações do modelo MILP para a variante com *Split* estão descritas logo a seguir:

$$l_i^v \leq e_k - t_{ijk}^v + \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (4.1)$$

$$a_j^v \geq b_k + t_{ijk}^v - \mathbf{e}_m \times (1 - x_{ijk}^v), \quad \forall (i, j) \in A; \quad \forall k \in K; \quad \forall v \in M \quad (4.2)$$

$$a_j^v \geq l_i^v + \sum_{k \in K} t_{ijk}^v - \mathbf{e}_m \times (1 - \bar{X}_{ij}^v), \quad \forall (i, j) \in A; \quad \forall v \in M \quad (4.3)$$

$$a_i^v + s_i \leq l_i^v, \quad \forall i \in N'; \quad \forall v \in M \quad (4.4)$$

$$S_i \leq a_i^v \leq E_i, \quad \forall i \in N'; \quad \forall v \in M \quad (4.5)$$

$$\bar{X}_{ij}^v \times D_{ij} = \sum_{k \in K} d_{ijk}^v, \quad \forall (i, j) \in A; \quad \forall v \in M \quad (4.6)$$

$$\sum_{(i,j) \in A} \sum_{v \in M} \bar{X}_{ij}^v \geq 1, \quad \forall j \in N \quad (4.7)$$

$$y_i^v \leq R_i \times \sum_{(i,j) \in A} \bar{X}_{ij}^v, \quad \forall i \in N'; \quad \forall v \in M \quad (4.8)$$

$$\sum_{v \in M} y_i^v = R_i, \quad \forall i \in N' \quad (4.9)$$

$$\sum_{i \in N'} y_i^v \leq C_v, \quad \forall v \in M \quad (4.10)$$

O modelo com *Split* exige a substituição da restrição (2.11) pela (4.6), a restrição (4.6) agora garante que, se o arco  $(i, j)$  é percorrido pelo veículo  $v$ , então o veículo  $v$  deve percorrer toda a distância  $D_{ij}$ . A consistência das variáveis  $l_i^v$  e  $a_i^v$  agora é assegurada respectivamente pelas restrições (4.1), (4.4) e (4.5), tais restrições substituem respectivamente as restrições (2.16), (2.19) e (2.20). Além disso, substituímos as restrições (2.6-2.9) pela restrição (4.7) a fim de permitir que o cliente possa ser atendido por mais de um veículo. As alterações realizadas nas restrições de tempo basicamente fixam os veículos  $v$  na desigualdade e aumentam a dimensão das variáveis  $l_i$  e  $a_i$ , tal alteração é necessária uma vez que existe a possibilidade de um cliente ser atendido por mais de um veículo em distintos instantes de tempo.

As restrições (4.8) até (4.10) tratam especificamente do *Split Delivery*, tais restrições devem ser adicionadas ao modelo original afim de tratar as peculiaridades do HGVRSP na sua variante com *Split*. Neste sentido as restrições (4.8) e (4.9) tratam da alocação das demandas, já a restrição (4.10) assegura que a soma das frações de demanda não excede a capacidade do veículo  $v$ .

## 4.2 Heurística Multi-Start Baseada em ILS

Esta seção descreve o algoritmo híbrido baseado na meta-heurística *Iterated Local Search* (ILS) utilizado para resolver o HGVRSP. A meta-heurística ILS é um método de trajetória que explora o espaço de soluções através da aplicação sucessiva de perturbações a ótimos locais [41]. Estes ótimos locais são obtidos a partir do algoritmo de busca local que, por sua vez, tem como ponto de partida uma solução inicial proveniente do procedimento construtivo ou de uma solução perturbada.

O algoritmo proposto, denominado MS-ILS, é uma heurística *Multi-Start* baseada na meta-heurística ILS, tendo como busca local o método *Randomized Variable Neighborhood Descent* (RVND). O RVND utiliza uma ordem aleatória de vizinhanças a cada chamada, essa ordem aleatória é uma vantagem, uma vez que não é necessário fazer experimentos para descobrir qual a melhor ordem [57, 55, 58]. Além da vantagem citada, esta abordagem do ILS com RVND, utilizada em [57], [49] e [52], também produz, em média, resultados melhores do que a versão com ordem determinística.

O MS-ILS proposto para esse trabalho é uma extensão do algoritmo apresentado em [48]. Algumas características são mantidas, porém devido ao grande número de restrições, novas características são adicionadas, como por exemplo a avaliação dos custos de carbono por veículo além da possibilidade do veículo realizar paradas antes do atendimento do cliente. Além disso, a inclusão da variante com restrições de entrega fracionada acarretaram profundas mudanças nas estruturas da solução assim como na dinâmica do algoritmo.

O pseudocódigo do MS-ILS proposto é descrito no Algoritmo 1. A heurística MS-ILS é formada por alguns procedimentos básicos, onde o procedimento **GeraSolucaoInicial** () (linha 6) é responsável pela construção de soluções iniciais. O segundo procedimento, **BuscaLocal** () (linha 7), é responsável por explorar o espaço de busca e encontrar bons ótimos locais, esta Busca Local usa o mecanismo de *Best improvement* <sup>1</sup>. O terceiro procedimento, **Perturba**() (linha 9), gera soluções modificadas de tal forma que o novo ótimo local obtido após a busca local seja diferente. O último elemento, identificado pelo procedimento **CriterioAceitacao**() (linha 11), é responsável por determinar se a solução proveniente da busca local será descartada ou utilizada na próxima iteração, o processo segue até que um número máximo de iterações seja alcançado (*maxIT* - linha 4).

O Algoritmo 1 ainda conta com recurso extra de iterações sem melhora (*MaxIterILS* - linha 8), dentro deste bloco de iterações sem melhora há um procedimento (linha 12)

---

<sup>1</sup> *Best improvement*: Seleciona a melhor solução aprimorante na vizinhança corrente.

responsável por armazenar as melhores soluções em um Conjunto Elite de Soluções denominado  $ES$ . Finalmente, numa fase posterior de intensificação, um modelo de cobertura de conjuntos é solucionado, alimentado com as soluções elite geradas no decorrer do método (linha 22).

---

**Algoritmo 1: MS-ILS ( $maxIT, maxIterILS$ )**


---

```

1 início
2    $S^* \leftarrow \emptyset$     $ES \leftarrow \emptyset$ 
3    $f(S^*) \leftarrow \infty$ 
4   para  $i \leftarrow 1$  até  $maxIT$  faça
5      $iterILS \leftarrow 0$ 
6      $S \leftarrow \text{GeraSolucaoInicial}()$ 
7      $S^{*'} \leftarrow \text{BuscaLocal}(S)$ 
8     enquanto  $iterILS \leq MaxIterILS$  faça
9        $S' \leftarrow \text{Perturba}(S^{*'})$ 
10       $S'' \leftarrow \text{BuscaLocal}(S')$ 
11      se  $f(S'') < f(S^{*'})$  então
12         $ES \leftarrow ES \cup \{S''\}$            ▷ Insere no conjunto de soluções elite
13         $S^{*'} \leftarrow S''$ 
14         $iterILS \leftarrow 0$ 
15      fim
16       $iterILS \leftarrow iterILS + 1$ 
17    fim
18    se  $f(S^{*'}) < f(S^*)$  então
19       $S^* \leftarrow S^{*'}$ 
20    fim
21  fim
22   $S^* \leftarrow \min(\text{setPartition}(ES), S^*)$ 
23  retorna  $S^*$                                ▷ Melhor solução
24 fim

```

---

Os procedimentos básicos do Algoritmo 1, componentes da solução, método de construção, busca local, perturbação e Cobertura de Conjuntos serão explorados em detalhes respectivamente nas Seções 4.2.1, 4.2.2, 4.2.3, 4.2.4 e Seção 4.3.

### 4.2.1 Representação da Solução

Pela natureza do problema, não é possível solucionar de maneira ótima instâncias grandes do problema em um baixo custo computacional, portanto um método heurístico foi considerado. Neste sentido, para representar uma solução  $S$  na heurística, foram utilizados dois vetores:  $R$  (com  $q$  linhas e um número variável de colunas) e  $Y$  (com  $q$  linhas e  $n$  colunas). Cada linha  $r$  do vetor  $R$  representará uma rota, a qual terá um veículo  $v$  associado. As colunas de cada uma dessas rotas indicarão uma permutação dos clientes,  $R = \{r_1 = [r_{10}, r_{12}, \dots, r_{1n}], r_2 = [r_{20}, r_{22}, \dots, r_{2n}], r_K = [r_{q0}, r_{q2}, \dots, r_{qn}]\}$ , sendo que as primeiras e últimas colunas conterão o depósito. No vetor  $Y$ , cada linha  $r$  estará associada a uma linha do vetor  $R$ . Cada uma das  $n$  colunas que compõem as linhas do vetor  $Y$

representará um cliente, sendo que  $Y_{rj}$  armazenará a fração de demanda do cliente  $j$  que é atendida pela rota  $r$ , caso contrário,  $Y_{rj} = 0$ .

A Figura 4.1 ilustra uma solução fictícia  $S$  para o HGVRSP. Na Figura 4.1 o cliente 7 é visitado pelas rotas  $r_1$  e  $r_2$ , além disso, temos a representação da solução utilizando os vetores  $R$  e  $Y$ . Para cada índice do vetor  $R$  temos armazenado o índice referente ao cliente visitado. Com estas representações, a informação acerca da quantidade de demanda suprida por cada rota, para um cliente, é acessada rapidamente.

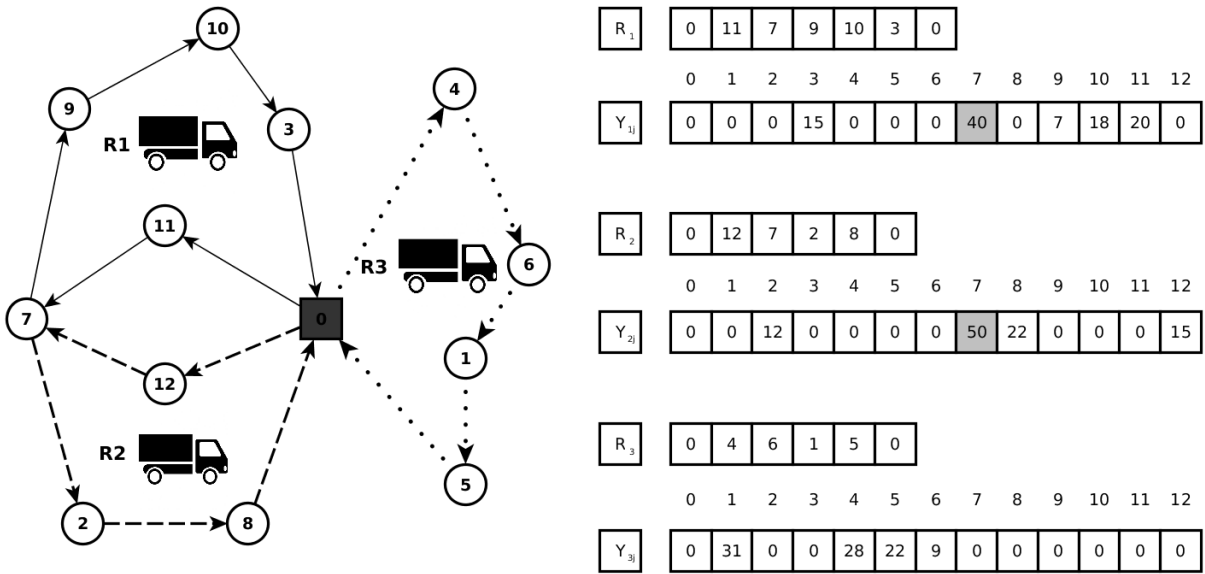


Figura 4.1: Solução com *Split-Delivery*.

Com o intuito de descrever a sequência de períodos ao longo de uma rota a matriz  $X_{ki}^v$  foi definida. As agendas para cada veículo  $v$  são representadas por uma matriz  $X_{ki}^v$  (com  $k$  linhas e um número variável de colunas), a matriz binária  $X_{ki}^v$  recebendo 1 indica que o arco  $(i, j)$  ( $i = R_v[i]$  e  $j = R_v[i + 1]$ ) é percorrido no período  $k$  pelo veículo  $v$ , o mesmo modelo de estrutura é aplicado a matriz  $d_{ki}^v$  e  $CO2_{ki}^v$ , porém, são respectivamente armazenados as frações de distância e  $CO_2$  pelo veículo  $v$  no arco  $(i, j)$ . Neste sentido, as informações sobre os períodos utilizados em cada rota  $v$  é acessado rapidamente.

A Figura 4.2 mostra um exemplo da codificação da mesma solução da Figura 4.1 com 12 clientes visitados por 3 veículos em um horizonte de 5 períodos de tempo. A rota realizada pelo veículo 1 percorre o arco  $(0, 11)$  apenas no período 0, ou seja, percorre do cliente  $i$  para  $i + 1$  no período  $k = 0$ . Supondo que o arco  $(0, 11)$  fosse percorrido no período 0 e 1 teríamos a variável  $X_{00}^1 = 1$  e  $X_{10}^1 = 1$ . Assim, obter os custos de carbono não é tão trivial uma vez que este custo depende diretamente do período em que o arco  $(i, j)$  é percorrido além do montante de cargas no veículo no arco corrente.

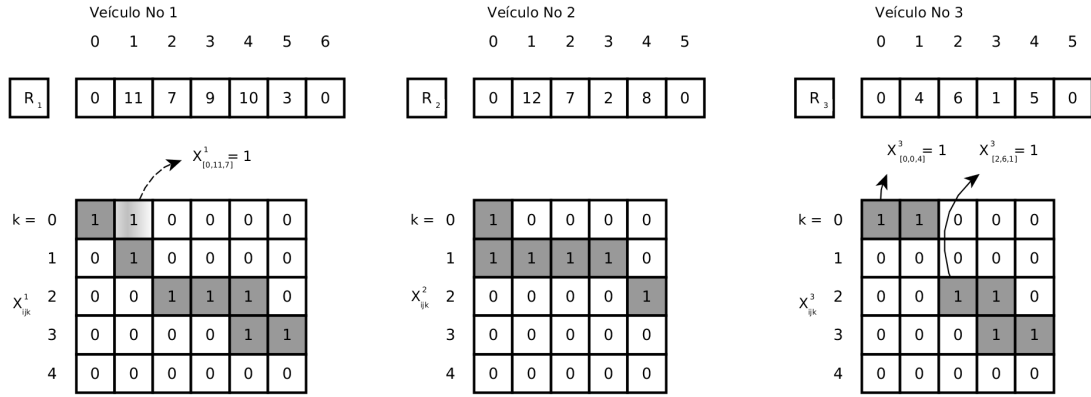


Figura 4.2: Agenda de acesso aos clientes.

O Algoritmo 2 calcula o custo de carbono gerado no arco  $(i, j)$  pelo veículo  $v$  partindo no instante  $l_i^v$  do cliente  $i$  até  $j$ . As primeiras linhas do Algoritmo 2 (linhas 2 - 5) calculam o custo de  $CO_2$  quando o veículo percorre o arco  $(i, j)$  estritamente no período  $k$ , além disso, adiciona-se o custo extra de  $CO_2$  causado pelas cargas presentes no veículo (linha 5). Se houver uma transição de um período  $k$  para  $k+1$  (linha 6), então um novo cálculo é realizado de forma proporcional às frações de distâncias percorridas nos respectivos períodos. O algoritmo segue calculando a fração de tempo ( $T1$ ) percorrido no período  $k$  e adicionando a uma *time-line* fictícia ( $T2$ ), a primeira etapa (linha 14) computa o custo de  $CO_2$  no período  $k$  e iterativamente calcula (linhas 15 - 37) os custos de carbono para os demais períodos até que não haja transição de períodos para percorrer o arco  $(i, j)$ .

Uma vez conhecidos os valores de  $d_{ki}^v$  para o arco  $(i, j)$  (onde  $k$  é o período ao qual o instante de tempo  $l_i^v$  pertence), o custo em carbono pode ser facilmente calculado e armazenado na estrutura  $CO2_{ki}^v$ . Também vale ressaltar que o montante de carga  $f_{ij}^v$  é previamente calculado em uma estrutura de dados auxiliar **DemandaDecrementada** (veja Seção 4.2.3.1). O mesmo procedimento do Algoritmo 2 pode ser usado para definir: tempo de chegada ( $a_j^v = T_{ij}$ ) no cliente  $j$ ; tempo de espera<sup>2</sup>  $TE_j$  antes de atender o cliente  $j$  ( $TE_j = a_j^v - S_j$  se  $a_j^v < S_j$ ,  $TE_j = 0$  c.c.) e tempo de saída do cliente  $j$  ( $l_j^v = a_j^v + s_j$ ). Após calcular o tempo de saída  $l_j^v$  pode-se ajustá-lo para adicionar um tempo de parada  $ST_j$  após o atendimento do cliente  $j$  ( $ST_j = |l_j^v - E_j|$  se  $l_j^v < E_j$ ,  $ST_j = 0$  c.c.). Uma vez calculado os custos de carbono para o arco  $(i, j)$ , o cálculo para os demais clientes de uma rota  $v$  demandaria de forma análoga avaliar o tempo de saída do cliente anterior. No HGVRSP, é fundamental pré-calculer os custos da rota  $v$  em  $CO_2$ , distância e tempo de trajeto. Esses custos são decisivos uma vez que os veículos apresentam restrições específicas de tempo de viagem e limite de distância baseado no consumo de combustível.

<sup>2</sup>Tempo de espera não contabiliza emissão de  $CO_2$  uma vez que o veículo encontra-se parado.

**Algoritmo 2:** CUSTO  $(i, j, l_i^v)$ 


---

```

1 início
2    $k \leftarrow \text{getMyPeriod}(l_i^v)$   $\triangleright$  Índice referente ao período do qual o tempo  $l_i^v$  pertence
3    $T_{ij} \leftarrow (D[i][j]/V[i][j][k])$   $\triangleright$  Tempo para percorrer de  $i$  a  $j$  com a  $k$ -ésima velocidade
4    $CO_2 \leftarrow c_v \times g_{ijk}^v \times D[i][j]$   $\triangleright$  Cálculo de  $CO_2$  causado pelo veículo vazio
5    $CO_2 \leftarrow CO_2 + c_v \times \varphi_v \times f_{ij}^v \times D_{ij}$   $\triangleright$  Adiciona emissão de  $CO_2$  causado pelas
   cargas do veículo no arco  $(i, j)$ 
6   se  $l_i^v + T_{ij} > e_k$  então
7     // Houve troca de períodos ao percorrer  $(i, j)$ , logo, há uma emissão de
      $CO_2$  no período  $k$  e  $k + 1$ . O cálculo de  $CO_2$  nos períodos  $k$  e  $k + 1$  deve
     ser proporcional às frações de distâncias percorridas no arco.
8
9      $T2 \leftarrow l_i^v$   $\triangleright$  Tempo de saída do cliente  $i$ 
10     $T_{ij} \leftarrow T1 \leftarrow (e_k - l_i^v)$   $\triangleright$  Fração de tempo percorrido no período corrente  $k$ 
11     $d \leftarrow T1 \times V[i][j][k]$   $\triangleright$  Fração de distância percorrida no período corrente  $k$ 
12     $d_{ki}^v \leftarrow \bar{d} \leftarrow d$   $\triangleright$  Armazena na estrutura auxiliar os valores de  $d$ 
13     $t_{ki}^v \leftarrow (d/V[i][j][k])$ 
14     $CO_2 \leftarrow c_v \times g_{ijk}^v \times d + (c_v \times \varphi_v \times f_{ij}^v \times D_{ij})$ 
15    para  $k' \leftarrow k + 1$  até  $M$  faça
16       $T2 \leftarrow T2 + T1$   $\triangleright$  Adiciona o tempo percorrido no período  $k$  na time-line
17       $d \leftarrow |D[i][j] - \bar{d}|$   $\triangleright$  Calcula a distância remanecente
18       $T1 \leftarrow d/V[i][j][k']$   $\triangleright$  Fração de tempo no período  $k'$ 
19      se  $T2 + T1 < e_{k'}$  então
20        // O tempo para percorrer os restante do arco foi pequeno o
        suficiente para não ter mudança de período
21
22         $T_{ij} \leftarrow T_{ij} + T1$   $\triangleright$  Tempo total para percorrer o arco  $(i, j)$ 
23         $d_{k'i}^v \leftarrow d$ 
24         $t_{k'i}^v \leftarrow (d/V[i][j][k'])$ 
25         $CO_2 \leftarrow CO_2 + (c_v \times g_{ijk'}^v \times d)$   $\triangleright$   $CO_2$  emitido no período  $k'$ 
26         $k' \leftarrow |M|$   $\triangleright$  Encerra o procedimento
27      fim
28    senão
29       $T1 \leftarrow |T2 - e_{k'}|$   $\triangleright$  Cálculo proporcional ao tempo no período  $k'$ 
30       $T_{ij} \leftarrow T_{ij} + T1$   $\triangleright$  Tempo total para percorrer o arco  $(i, j)$ 
31       $d \leftarrow T1 * V[i][j][k']$ 
32       $\bar{d} \leftarrow \bar{d} + d$ 
33       $d_{k'i}^v \leftarrow d$ 
34       $t_{k'i}^v \leftarrow (d/V[i][j][k'])$ 
35       $CO_2 \leftarrow CO_2 + (c_v \times g_{ijk'}^v \times d)$   $\triangleright$   $CO_2$  emitido no período  $k'$ , exige uma
      nova iteração para calcular o  $CO_2$  no período  $k' + 1$ 
36    fim
37  fim
38 fim
39 retorna  $CO_2$ 
40 fim

```

---

### 4.2.2 Método Construtivo

O procedimento de construção da solução inicial foi adaptado de métodos existentes na literatura. Uma estratégia de inserção foi adotada, a Estratégia de Inserção Paralela (EIP), onde todas as rotas são consideradas, paralelamente, durante o processo construtivo. Com a finalidade de gerar soluções iniciais diversificadas, dois critérios de inserção foram utilizados: Critério de Inserção mais Barata Viável (CIMBV) e o Critério de Inserção mais Próxima Viável (CIMPV).

O pseudocódigo **GeraSolucaoInicial()** é apresentado no Algoritmo 3. Inicialmente, a Lista de Candidatos (LC) é preenchida com todos os clientes (linha 2). Em seguida, cada rota  $r'$  é preenchida com um cliente  $h$  selecionado, aleatoriamente, da LC (linhas 4 - 7). Também aleatoriamente é selecionado o critério de inserção (linha 8). A solução inicial é gerada pela EIP (linha 9) usando o critério previamente selecionado.

---

**Algoritmo 3:** GERASOLUÇÃOINICIAL ( $v$ )

---

```

1 início
2   Inicializa LC ▷ lista com todos os clientes
3    $s \leftarrow \{r^1, \dots, r^v\}$  ▷ conjunto composto por  $v$  rotas vazias
4   para  $r' \leftarrow 1$  até  $v$  faça
5      $s^{r'} \leftarrow h \in LC$  ▷ selecionado aleatoriamente
6     Atualize a LC ▷  $LC \leftarrow LC \setminus \{h\}$ 
7   fim
8   CritérioInserção  $\leftarrow$  CIMBVM ou CIMPV ▷ selecionado aleatoriamente
9    $s \leftarrow$  EIP( $s, r, LC, \text{CritérioInserção}$ )
10  retorna  $s$ 
11 fim

```

---

No critério CIMBV, o custo de inserção de um cliente não roteado  $h \in LC$  em uma determinada rota, associada a um veículo  $r$ , é expresso pela Equação (4.11), na qual a função  $g^r(h)$  representa o custo de inserção. O valor de  $g^r(h)$  é calculado pela soma de dois termos, o primeiro termo computa o custo de inserção do cliente  $h$  entre cada par de clientes adjacentes  $i$  e  $j$  da rota, enquanto o segundo define a penalização usada para desestimular inserções tardias de clientes com alto custo de carbono. O fator  $\gamma$  define o peso do custo de ida e volta ao depósito.

$$g^r(h) = (c_{ih}^r + c_{hj}^r - c_{ij}^r) - \gamma(c_{0h}^r + c_{h0}^r) \quad (4.11)$$



O critério CIMPV calcula o custo em carbono entre um cliente  $h \in LC$  e todos os clientes que já tenham sido incluídos na solução parcial, ou seja  $g^r(h) = c_{ih}^r$  Equação (4.12). Assume-se que a inserção de  $h$  é sempre realizada após  $i$ .

$$g^r(h) = c_{ih}^r \quad (4.12)$$

Nos dois critérios adotados o cliente  $i_{min}$  inserido é aquele associado ao menor custo, isto é,  $i_{min} \leftarrow \operatorname{argmin}\{ g^r(h) \mid h \in LC \}$ .

---

**Algoritmo 4:** EIP (s, r, LC, CritérioInserção)

---

```

1 início
2    $\gamma \leftarrow 0$  ▷ inicializa  $\gamma$ 
3   se CritérioInserção == CIMBVM então
4      $\gamma \leftarrow$  valor aleatório dentro de um dado intervalo
5   fim
6   enquanto ( $LC \neq \emptyset$  e pelo menos um cliente  $i \in LC$  pode ser adicionado a s) faça
7     Calcule o custo  $g^r(h)$  para cada  $i \in LC$  e para cada rota com veículo  $r$ 
8      $i' \leftarrow \operatorname{argmin}\{ g^r(h) \mid h \in LC \text{ e veículo } r \}$ 
9      $r' \leftarrow$  rota associada ao  $i'$ 
10     $s^{r'} \leftarrow s^{r'} \cup \{i'\}$ 
11    Atualize LC ▷  $LC \leftarrow LC \setminus \{i\}$ 
12  fim
13  se  $|LC| > 0$  então
14    Adicione uma nova rota com um veículo fictício, com os custos elevados
15    e que comporte todos os clientes
16    Vá para linha 6
17  fim
18  retorna s
19 fim
```

---

Na Estratégia de Inserção Paralela (EIP), cujo pseudocódigo é apresentado no Algoritmo 4, todas as rotas são consideradas a cada iteração. Se o critério de inserção selecionado for o CIMBVM, um valor aleatório dentre  $\{0.00, 0.05, 0.10, \dots, 1.65, 1.70\}$  é selecionado para  $\gamma$ . Estes valores foram estipulados empiricamente. Em seguida, enquanto a LC não estiver vazia e existir pelo menos um cliente  $i \in LC$  que possa ser adicionado à solução parcial corrente, sem violar nenhuma restrição (linhas 6 - 12), um cliente  $i'$  é adicionado à rota  $r'$  de menor custo de inserção segundo o critério de inserção selecionado. Se ao final das iterações ainda restar algum(s) cliente(s), uma nova rota é associada à um novo veículo fictício, com os custos muito superiores aos custos existentes e sem restrição de janela de tempo em relação ao atendimento dos cliente (linha 14).

Ao avaliar as restrições de janela de tempo onde o veículo adicionado não possui restrições de atendimento a nenhum cliente, então este veículo faz com que a solução seja inviável. Entretanto, os valores dos custos deste novo veículo são elevados, então na fase de busca local qualquer movimento que retire um cliente da rota associada a esse veículo será atrativo. Logo, nenhum movimento que insira clientes nessa rota será vantajoso. Após a criação da nova rota, na linha 16 o procedimento é reiniciado a partir da linha 6.

### 4.2.3 Busca Local

A busca local é feita pelo procedimento RVND [48]. O RVND é uma variação do método *Variable Neighborhood Descent* - VND [24], que ao invés de usar uma ordem pré-definida de vizinhanças para explorar o espaço de soluções, utiliza uma ordem aleatória a cada chamada. Mais especificamente, sempre que em uma determinada vizinhança não for possível melhorar a solução corrente, o RVND seleciona, aleatoriamente, outra vizinhança para continuar a busca pelo espaço de soluções.

---

**Algoritmo 5: BUSCALOCAL (s)**


---

```

1  início
2      Atualize as EDAs                                ▷ Veja Seção 4.2.3.1
3      Inicialize a Lista de Vizinhanças (LV) Inter-Rota
4      enquanto (LV ≠ ∅) faça
5          Escolha uma vizinhança  $\mathcal{N} \in LV$  aleatoriamente
6          Encontre o melhor vizinho  $s'$  de  $\mathcal{N}(s)$ 
7          se ( $f(s') < f(s)$ ) então
8               $s \leftarrow s'$ 
9               $s \leftarrow BuscaIntrarota(s)$ 
10             Reinicialize a LV
11         fim
12     senão
13         Remova  $\mathcal{N}$  da LV
14     fim
15     Atualize as EDAs
16 fim
17 retorna s
18 fim

```

---

O pseudocódigo do procedimento RVND é apresentado no Algoritmo 5. Seja LV uma Lista de Vizinhanças inter-rotas. A LV é inicializada com vizinhanças associadas aos movimentos descritos na Seção 4.2.3.3 (linha 3). A cada iteração (linhas 4 - 16) uma vizinhança  $\mathcal{N} \in LV$  é selecionada aleatoriamente (linha 5) e o melhor movimento viável é determinado (linha 6). Em caso de melhora da solução corrente, uma busca local intra-rota é executada (linha 9 - veja Algoritmo 6), a frota é atualizada e a LV é reinicializada com todas as vizinhanças (linhas 7 - 11). Caso contrário  $\mathcal{N}$  é removida da LV (linha 13). Um conjunto de Estruturas de Dados Auxiliares (EDA) (Seção 4.2.3.1) é atualizado no início do método (linha 2) e sempre que uma vizinhança é executada (linha 15).

A busca local em todas as vizinhanças é realizada de forma exaustiva, considerando apenas as soluções viáveis. A princípio, verifica-se a viabilidade das cargas e consumo de combustível para todas as rotas. Neste trabalho são aceitas somente soluções que melhoram a solução corrente, porém são permitidas soluções inviáveis quando avaliado as janelas de tempo, estas soluções inviáveis são penalizadas de acordo com um fator que indica o quão distante os clientes estão sendo atendidos em relação à sua Janela de Tempo. A Seção 4.2.3.2 explica como analisar a viabilidade em tempo constante.

---

**Algoritmo 6: BUSCAINTRAROTA ( $s$ )**


---

```

1  início
2  | Inicialize a Lista de Vizinhanças Intrarotas ( $LV'$ )
3  | enquanto ( $LV' \neq \emptyset$ ) faça
4  | | Escolha uma vizinhança  $\mathcal{N}' \in LV'$  aleatoriamente
5  | | Encontre o melhor vizinho  $s'$  de  $\mathcal{N}'(s)$ 
6  | | se ( $f(s') < f(s)$ ) então
7  | | |  $s \leftarrow s'$ 
8  | | fim
9  | | senão
10 | | | Remova  $\mathcal{N}'$  da  $LV'$ 
11 | | fim
12 | fim
13 | retorna  $s$ 
14 fim

```

---

#### 4.2.3.1 Estruturas de Dados Auxiliares (EDA)

Com a finalidade de acelerar a busca e evitar a avaliação desnecessária de movimentos, algumas Estruturas de Dados Auxiliares (EDA) foram desenvolvidas. As seguintes matrizes armazenam informações úteis para cada rota.

- EDAs para cargas

**SomaEntrega[ ]** : armazena a soma das demandas de uma rota  $r$ . Por exemplo, **SomaEntrega[ 2 ]** = 300 indica que a soma das demandas de todos os clientes da rota 2 corresponde a 300.

**EntregaMin[ ]** : demanda mínima da rota  $r$ .

**EntregaMax[ ]** : demanda máxima da rota  $r$ .

**EntregaMinAdj[ ]** : soma mínima das demandas de dois clientes adjacentes da rota  $r$ . Por exemplo, **EntregaMinAdj[ 4 ]** = 10 indica que a menor soma das demandas de dois clientes adjacentes da rota 4 é 10.

**EntregaMaxAdj[ ]** : soma máxima das demandas de dois clientes adjacentes da rota  $r$ .

**DemandaAcumulada[ ][ ]** : matriz tendo como argumentos uma rota  $r$  e uma posição  $pos$  na sequência de visita, que retorna a carga da demanda acumulada até a posição  $pos$  da rota  $r$ . Por exemplo, **DemandaAcumulada[ 2 ][ 4 ]** = 78 significa que a soma dos quatro primeiros clientes da rota 2 totaliza 78.

**DemandaDecrementada[ ][ ]** : semelhante a estrutura de Demanda Acumulada, porém retorna a carga da demanda decrementada ao longo de uma rota  $r$ . Esta estrutura é essencial uma vez que a demanda acumulada no arco  $(i, j)$  interfere nos custos da função objetivo do problema.

- EDAs para avaliar Janelas de Tempo

**Duração[ ][ ][ ]** : matriz tendo como argumentos uma rota  $r$ , uma posição inicial  $ini$  e uma posição final  $fim$  de uma sequência de visita, que retorna a duração da rota, ou seja, soma do tempo de viagem e do tempo de serviço entre as posições  $ini$  e  $fim$  da sequência de visita da rota.

**Earliest[ ][ ][ ]** : matriz que armazena para uma sequência de visitas da rota  $r$  começando na posição inicial  $ini$  e terminando em uma posição final  $fim$ , o instante de tempo a partir da qual é possível iniciar a visita da sequência de clientes para completá-la de modo viável.

**Latest[ ][ ][ ]** : armazena o último instante de tempo que é possível iniciar a visita da sequência dada.

**PenalidadeJT[ ][ ][ ]** : armazena a quantidade de tempo passado após o fim da janela de tempo em que foi possível iniciar o atendimento da sequência de visitas.

Para atualizar as informações das EDAs deve-se levar em consideração apenas as rotas que foram modificadas. Seja  $\bar{n}$  o número total de clientes nas rotas modificadas, para cada rota modificada, uma verificação é efetuada ao longo da rota para atualizar os valores correspondentes das EDAs. Desta forma, a complexidade computacional é  $\mathcal{O}(\bar{n})$ . Ao aplicar as vizinhanças inter-rotas a complexidade computacional é elevada para  $\mathcal{O}(\bar{r} \mid \bar{n} \mid)$ , sendo  $\bar{r}$  o número de rotas modificadas.

### 4.2.3.2 Tratamento das Inviabilidades usando as EDAs

Uma parte importante do tempo computacional utilizado para resolver um problema é gasto para checar a viabilidade da solução ou para se computar o seu custo após a realização de um movimento. Nesta seção são descritos as EDAs envolvidas no problema para checar custo e viabilidade em tempo constante.

As EDAs exploram o fato das vizinhanças adotadas utilizarem mecanismos de troca e realocação de clientes, que podem ser vistos como uma recombinação de sequências de clientes na solução [31, 28, 61].

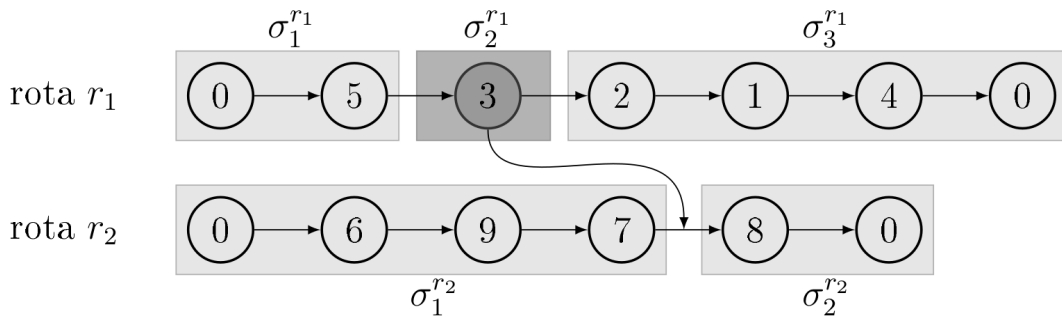


Figura 4.3: Exemplo de Recombinação de Sequências.

A Figura 4.3 ilustra o movimento de realocação do cliente 3 da rota  $r_1$  para depois do cliente 7 da rota  $r_2$  e pode ser visto como uma recombinação das sequências. Na rota  $r_1$  temos as sequências  $\sigma_1^{r1}$ ,  $\sigma_2^{r1}$  e  $\sigma_3^{r1}$  e na rota  $r_2$  as sequências  $\sigma_1^{r2}$  e  $\sigma_2^{r2}$ . A nova solução é formada pelas rotas  $r'_1$  e  $r'_2$ , onde  $r'_1 = \sigma_1^{r1} \oplus \sigma_3^{r1}$  e  $r'_2 = \sigma_1^{r2} \oplus \sigma_2^{r1} \oplus \sigma_2^{r2}$ , ou seja, é executada uma concatenação das sequências, como mostrado na Figura 4.4.

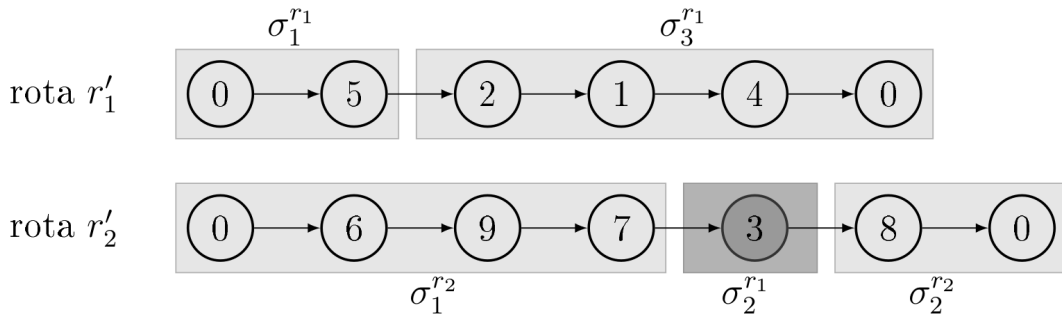


Figura 4.4: Sequências Recombinadas para Execução de Movimentos.

Para que a avaliação dos movimentos possa ser efetuada em tempo constante, um pré-processamento das rotas da solução deve ser efetuado. Neste sentido, os seguintes procedimentos, utilizados em [48], foram adaptados para melhorar o desempenho da busca local:

- **Inicializa( $\sigma$ )** : Inicializa as EDAs da sequência  $\sigma$ , onde cada subsequência possui apenas um cliente.
- **Forward( $\sigma$ )** : Para cada subsequência, calcula os valores para o próximo cliente da sequência.
- **Backward( $\sigma$ )** : Computa os valores das EDAs para cada subsequência em ordem reversa.
- **Avalia( $\sigma_1, \sigma_2$ )** : Avalia o custo e a viabilidade da concatenação da sequência  $\sigma_1$  com  $\sigma_2$  ( $\sigma_1 \oplus \sigma_2$ ).

Neste trabalho, permite-se que o cliente seja atendido fora da sua janela de tempo, no entanto uma penalidade é dada para esse atendimento, caso seja efetuado após o término da janela de tempo. O atendimento antes da janela de tempo não sofre penalização, sendo apenas computado o tempo de espera (TE) até o início da janela de tempo para se começar o atendimento. A Figura 4.5 exemplifica a modelo de penalização adotado.

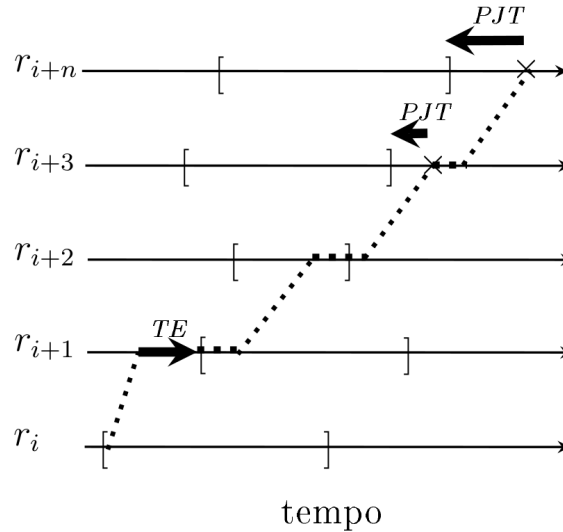


Figura 4.5: Penalização da Janela de Tempo.

A Figura 4.5 descreve o tempo gasto no atendimento a determinados clientes de uma rota  $r$ . As linhas pontilhadas na vertical indicam o tempo de viagem de um cliente a outro, enquanto que na horizontal indicam o tempo de serviço. Na Figura 4.5, o veículo  $r$  chega antes do início da janela de tempo do cliente  $i + 1$ , logo, um tempo espera **TE** se faz necessário para iniciar o atendimento. Por outro lado, o atendimento do cliente  $i + 3$  e  $i + n$  acontece após o término da suas janelas de tempo, portanto uma penalização por violação das janelas de tempo **PJT** é dada a esses clientes nessa rota. Este modelo de penalização da janela de tempo foi adotado com sucesso nos trabalhos de [45] e [62].

Os cálculos dessas penalidades utilizados na verificação da solução são efetuados como segue. Para cada subsequência qualquer de visitas  $\sigma$ , as EDAs são computadas utilizando os procedimentos **Inicializa()**, **Backward()**, **Forward()** e **Avalia()**. Para uma sequência  $\sigma_0$  contendo um único vértice  $i$  as EDAs são inicializadas com **Duração**( $\sigma_0$ ) =  $S_i$ , **Earliest**( $\sigma_0$ ) =  $E_i$ , **Latest**( $\sigma_0$ ) =  $s_i$  e **PenalidadeJT**( $\sigma_0$ ) = 0. A concatenação de sequências permite calcular estes valores para toda a rota. Sejam  $\sigma_1 = (\sigma_{1(i)}, \dots, \sigma_{1(j)})$  e  $\sigma_2 = (\sigma_{2(i)}, \dots, \sigma_{2(j)})$  duas subsequências de visitas. As EDAs para as subsequências concatenadas  $\sigma_1 \oplus \sigma_2$  são calculadas da seguinte forma:

$$Duracao(\sigma_1 \oplus \sigma_2) = Duracao(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} + Duracao(\sigma_2) + \Delta_{TE} \quad (4.13)$$

$$Earliest(\sigma_1 \oplus \sigma_2) = \max\{Earliest(\sigma_2) - \Delta, Earliest(\sigma_2)\} - \Delta_{TE} \quad (4.14)$$

$$Latest(\sigma_1 \oplus \sigma_2) = \min\{Latest(\sigma_2) - \Delta, Latest(\sigma_1)\} + \Delta_{PJT} \quad (4.15)$$

$$PenalidadeJT(\sigma_1 \oplus \sigma_2) = PenalidadeJT(\sigma_1) + PenalidadeJT(\sigma_2) + \Delta_{PJT} \quad (4.16)$$

onde,

$$\Delta = Duracao(\sigma_1) - PenalidadeJT(\sigma_1) + d_{\sigma_1(|\sigma_1|)\sigma_2(1)} \quad (4.17)$$

$$\Delta_{TE} = \max\{Earliest(\sigma_2) - \Delta - Latest(\sigma_1), 0\} \quad (4.18)$$

$$\Delta_{PJT} = \max\{Earliest(\sigma_1) + \Delta - Latest(\sigma_2), 0\} \quad (4.19)$$

Por meio dos procedimentos **Inicializa()**, **Backward()**, **Forward()** e **Avalia()** e das Equações (4.13) - (4.19) é possível realizar um pré-processamento da solução, para obter o custo em tempo de uma rota  $r$ , pela Equação (4.20)

$$F(r) = c_{(0)(|r|)}^u + \omega \times PenalidadeJT(r) \quad (4.20)$$

Na Equação (4.20), o valor  $\omega$  indica o peso dado para a PJT. O termo  $c_{(0)(|r|)}^u$ , indica o custo de carbono do cliente na posição 0 até o cliente da posição  $|r|$ , utilizando-se o veículo  $u$ . Já o termo  $\omega \times PenalidadeJT(r)$ , representa o valor da violação  $PJT$  vezes um determinado peso para a rota  $r$ . Caso todos os clientes sejam atendidos dentro da janela de tempo o valor da  $PenalidadeJT(r)$  é igual a zero. Portanto, este segundo termo da equação será igual a zero, retornando como custo o valor  $c_{(0)(|r|)}^u$ .

### 4.2.3.3 Vizinhos Inter-rotas

Com o intuito de reduzir os custos de transporte entre clientes de rotas distintas, foram implementadas algumas estruturas de vizinhança inter-rotas. São elas:

- **Shift(1,0)**: Um cliente é removido de uma rota e inserido em outra. Na Figura 4.6(b), o cliente 7 é removido da rota  $r_1$  e inserido na rota  $r_2$ . Utilizando as EDAs, se a operação  $\mathbf{EntregaMin}[r_1] + \mathbf{SomaEntrega}[r_2] > Q_{v(r_2)}$ , com  $Q_{v(r_2)}$  representando a capacidade do veículo associado à rota  $r_2$ , então a transferência de qualquer cliente de  $r_1$  para  $r_2$  implica em uma solução inviável.
- **Swap(1,1)**: Permuta dois clientes de rotas distintas. Na Figura 4.6(c), o cliente 2 da rota  $r_1$  é trocado com o cliente 6 da rota  $r_2$ . Para evitar a avaliação de movimentos inviáveis deve-se efetuar as seguintes verificações para o segmento de rota  $r_1$ :  $\mathbf{EntregaMin}[r_2] - \mathbf{EntregaMax}[r_1] + \mathbf{SomaEntrega}[r_1] \leq Q_{v(r_1)}$ ;  $R_j + \mathbf{SomaEntrega}[r_1] - \mathbf{EntregaMax}[r_1] \leq Q_{v(r_1)}$ , assim como para a rota  $r_2$ :  $\mathbf{EntregaMin}[r_1] - \mathbf{EntregaMax}[r_2] + \mathbf{SomaEntrega}[r_2] \leq Q_{v(r_2)}$ ;  $R_i + \mathbf{SomaEntrega}[r_2] - \mathbf{EntregaMax}[r_2] \leq Q_{v(r_2)}$ . As cargas dos veículos das duas rotas são examinadas de modo similar. Por exemplo, no caso de  $r_2$ , todos os clientes situados antes da posição que  $j$  foi encontrado (agora substituído por  $i$ ), têm suas cargas adicionadas por  $R_i$  e subtraídas por  $R_j$ .
- **Shift(2,0)**: Dois clientes consecutivos  $i$  e  $j$  são transferidos de uma rota  $r_1$  para outra  $r_2$ . Na Figura 4.6(d) os clientes adjacentes 7 e 11 são movidos de uma rota para outra. Antes de iniciar a avaliação dos clientes a serem transferidos de  $r_1$  para  $r_2$  devemos verificar se a seguinte condição é satisfeita:  $\mathbf{EntregaMinAdj}[r_1] + \mathbf{SomaEntrega}[r_2] \leq Q_{v(r_2)}$ . A carga do veículo na rota  $r_2$  é feita usando-se uma abordagem similar à adotada na vizinhança **Shift(1,0)**. Todos os clientes localizados antes da posição de inserção em  $r_2$  têm suas cargas somadas em  $R_i + R_j$ . A nova carga do arco  $(i, j)$  também deve ser verificada.
- **Swap(2,1)**: Dois clientes consecutivos  $i$  e  $j$  pertencentes a uma rota  $r_1$  são permutados com um cliente  $i'$  de outra rota  $r_2$ . Na Figura 4.6(e) os clientes adjacentes 7 e 11 de uma rota são trocados com o cliente 2 de outra rota. A avaliação de alguns movimentos inviáveis é evitada, realizando-se os seguintes testes:  $\mathbf{EntregaMinAdj}[r_1] + \mathbf{EntregaMax}[r_2] \leq Q_{v(r_2)}$ . A carga é verificada de modo similar as abordagens usadas nas vizinhanças **Shift(2,0)** e **Swap(1,1)**.



- **Swap(2,2)**: Dois clientes consecutivos  $i$  e  $j$  pertencentes a uma rota  $r_1$  são permutados com outros dois clientes consecutivos  $i'$  e  $j'$  de outra rota  $r_2$ . Na Figura 4.6(f) os clientes adjacentes 6 e 7 de uma rota são trocados com os clientes adjacentes 1 e 2 de outra rota. A avaliação de alguns movimentos inviáveis é evitada, realizando-se os seguintes testes:  $\mathbf{EntregaMinAdj}[r_1] + \mathbf{EntregaMax}[r_2] + \mathbf{SomaEntrega}[r_2] \leq Q_{v(r_2)}$ . A carga é verificada de modo similar a abordagem usada na vizinhança **Swap(2,1)**.
- **Cross**: O arco entre os clientes adjacentes  $i$  e  $i + 1$ , pertencentes a uma rota  $r_1$ , e o arco entre os clientes adjacentes  $j$  e  $j + 1$  de outra rota  $r_2$ , são removidos. Em seguida, para se inserir novos arcos, os clientes  $i$  e  $j + 1$  são conectados, assim como os clientes  $j$  e  $i + 1$ . Na Figura 4.6(g) os arcos  $(1,2)$  e  $(6,7)$  são removidos e os arcos  $(6,2)$  e  $(1,7)$  são inseridos. O procedimento para testar a carga do veículo de cada rota é feita como segue. A carga inicial ( $\mathcal{L}_0$ ) e final ( $\mathcal{L}_f$ ) dos veículos das duas rotas é calculado utilizando-se as EDAs **SomaEntrega** e **EntregaAcumulada**. Se os valores de ( $\mathcal{L}_0$ ) e ( $\mathcal{L}_f$ ) não excederem a capacidade do veículo  $Q_v$ , então as cargas intermediárias são verificadas pela seguinte expressão:  $\mathcal{L}_i = \mathcal{L}_{i-1} - R_i$ . Consequentemente, se  $\mathcal{L}_i$  ultrapassar  $Q_v$ , o movimento é inviável.

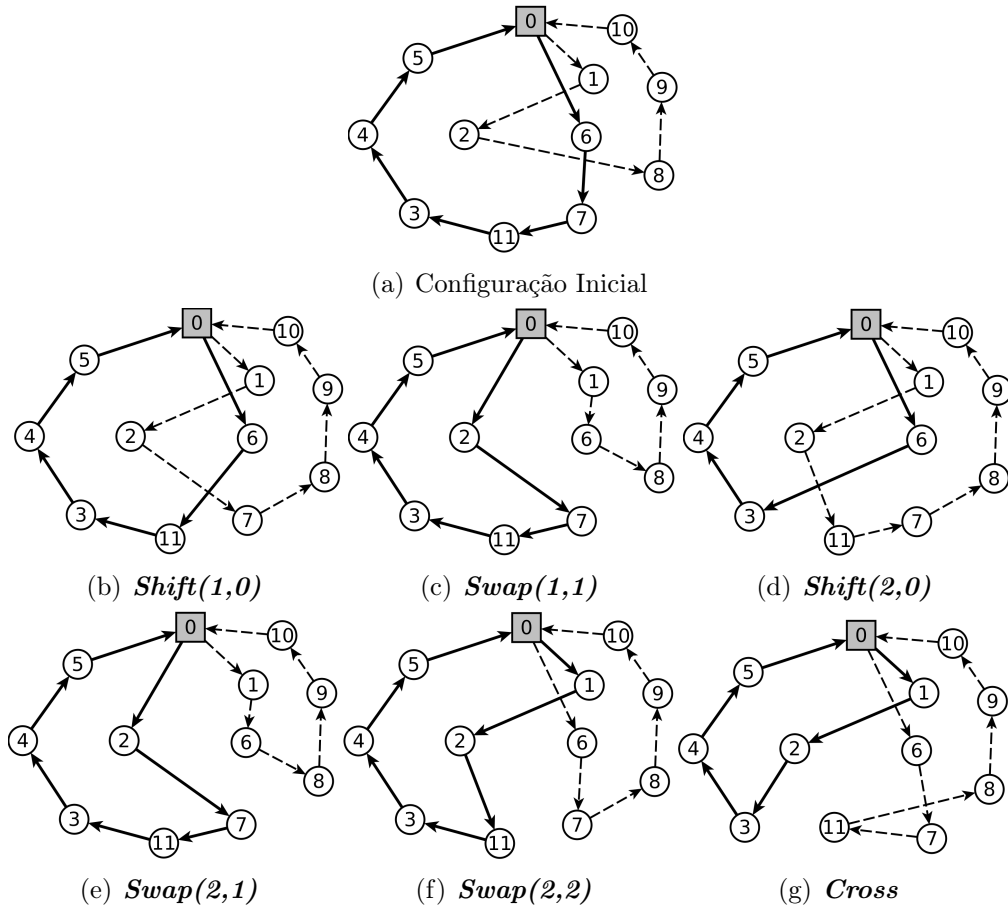


Figura 4.6: Vizinhanças Inter-Rotas [48].

#### 4.2.3.4 Vizinhanças Intrarotas

Para melhorar os custos de transporte em cada rota, quatro estruturas de vizinhança intrarota foram aplicadas. São elas:

- **Shift**: Um cliente é removido e reinserido em outra posição da rota. Na Figura 4.7(b) o cliente 3 é realocado em outra posição na rota.
- **Swap**: Similar ao movimento **Swap(1,1)**, consiste na permutação entre dois clientes. Na Figura 4.7(e) os clientes 2 e 6 são trocados.
- **2-opt**: Dois arcos não adjacentes  $(i_1, j_1)$  e  $(i_2, j_2)$  são removidos. Em seguida, dois novos arcos  $(i_1, i_2)$  e  $(j_1, j_2)$  são inseridos. Na Figura 4.7(d) os arcos  $(2,3)$  e  $(5,6)$  são removidos para a criação de dois novos arcos  $(2,5)$  e  $(3,6)$ .
- **Or-opt2**: Dois clientes consecutivos são removidos e reinseridos em outra posição da rota. Na Figura 4.7(c) os clientes adjacentes 2 e 3 são realocados em outra posição.

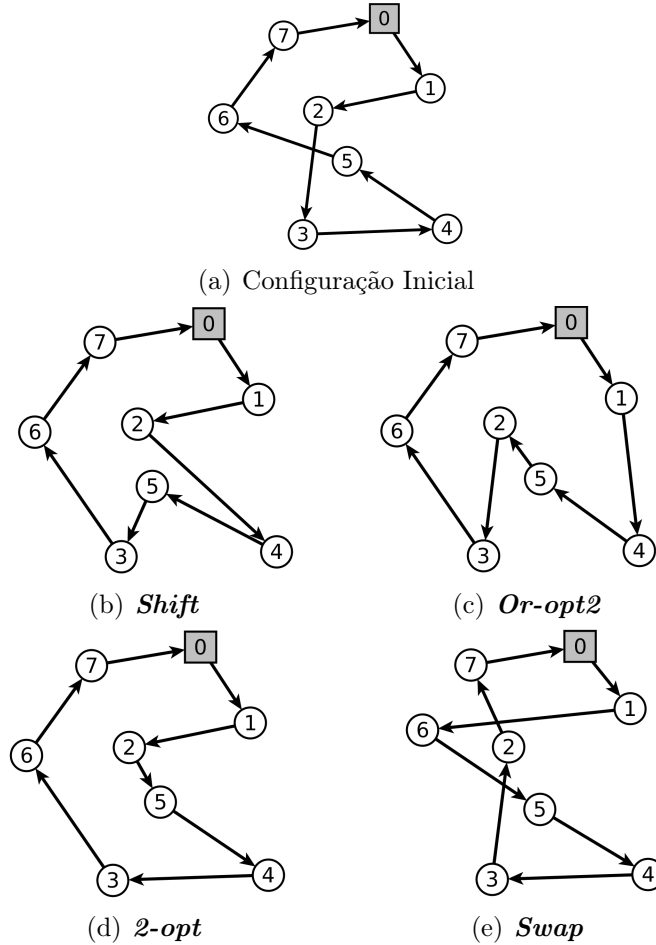


Figura 4.7: Vizinhanças Intrarotas [48].

#### 4.2.3.5 Vizinhanças para o *Split Delivery*

Por fim, as vizinhanças *RouteAddition* e *k-Split* foram adotadas para a variante com entrega fracionada. As estruturas de vizinhanças do tipo inter-rotas foram baseadas no trabalho proposto por [52], as estruturas estão descritas a seguir:

- ***RouteAddition***: Cria uma nova rota  $r_4$  para um dado cliente  $i$  que aparece em pelo menos duas rotas,  $r_1$  e  $r_2$ . Na Figura 4.8, o cliente 7 é removido das rotas originais e uma nova rota  $r_4$  é criada, associada a um veículo de capacidade suficiente, selecionado aleatoriamente.

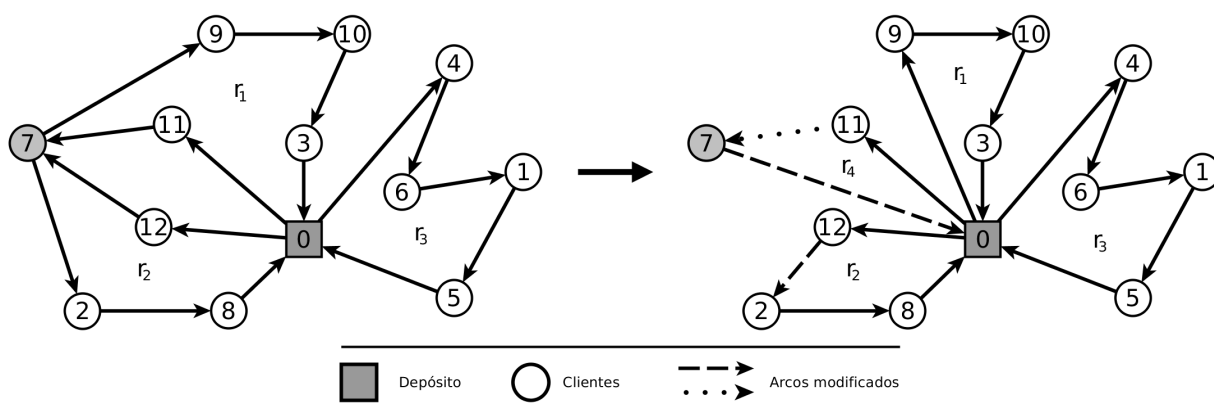


Figura 4.8: Vizinhança inter-roteira *RouteAddition*.

- ***k-Split***: Divide a carga de um cliente entre vários veículos. Remove um cliente das rotas em que ele aparece e realoca-o de tal modo que ele seja inserido na melhor posição, em cada rota que possuir capacidade residual para atender parte da sua demanda. Na Figura 4.9, o cliente 7 da rota  $r_4$  tem a carga dividida com a rota  $r_2$ .

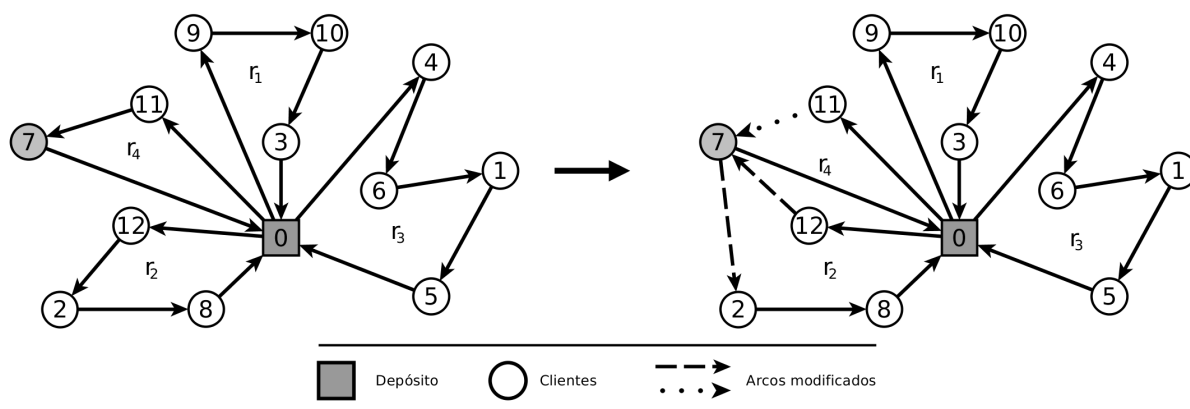


Figura 4.9: Vizinhança inter-roteira *k-Split*.

#### 4.2.3.6 Vizinhanças para o HGVRSP

Outras vizinhanças são propostas nesse trabalho especificamente para o HGVRSP, as vizinhanças propostas consistem em alterar a sequência de períodos durante uma rota ou aplicar frações de paradas nos arcos  $(i, j)$ .

- ***addIdleTime(i, j, v)***: Essa vizinhança adiciona um tempo de parada em um arco  $(i, j)$  em que ocorre transição de períodos. Seja um arco  $(i, j)$  percorrido em um período  $k$  e  $k + 1$  por um veículo  $v$ , um tempo de saída  $l_i^v$ , tempo de chegada  $a_j^v$  e tempo de término do período  $e_k$ , assim, pode-se alterar o tempo de saída  $l_i^v \leftarrow b_{k+1}$ , essa alteração adiciona um tempo de parada após a saída do cliente  $i$  e garante que parte do arco  $(i, j)$  não será percorrido no período  $k$ . Após essa alteração deve-se recalculer os tempos de chegada e custos de carbono via Algoritmo 2 para todos os clientes subsequentes a  $i$  da rota  $v$  afim de verificar se houve redução de custo. A vizinhança ***addIdleTime(i, j, v)*** é viável uma vez que, é permitido realizar uma parada no arco além de abrir a possibilidade para evitar percorrer parte do arco em um período  $k$  com alto custo de carbono.
- ***decreaseIdleTime(i, j, v)***: Vizinhança análoga a ***addIdleTime(i, j, v)***, porém seleciona-se um arco  $(i, j)$  que seja percorrido em um período  $k$  e  $k + 1$  por um veículo  $v$ , para reduzir o tempo de saída  $l_i^v$  deve-se reavaliar o tempo de saída do cliente  $i$  (**se**  $l_i^v > (a_i^v + s_i)$  **então**  $l_i^v \leftarrow a_i^v + s_i$ ) e recalculer os tempos de chegada e custos de carbono via Algoritmo 2 para todos os clientes subsequentes a  $i$ . Essa redução de tempo pode garantir que parte do arco  $(i, j)$  seja percorrido por mais tempo no período  $k$  e reduzir o tempo percorrido no período  $k + 1$ . Da mesma forma que adicionamos tempo de espera pela vizinhança ***addIdleTime(i, j, v)*** precisamos de uma vizinhança para reduzir os tempos em arcos com transição de períodos com alto custo de carbono.
- ***addIdleTimeDepot(0, j)***: Adiciona um tempo de parada após a saída do depósito pelo veículo  $v$ , neste sentido, aplica-se um considerável tempo de parada na variável  $l_0^v$ , i.e.,  $l_0^v \leftarrow b_{k+1}$ . Neste sentido, a rota  $v$  inicia-se a partir do próximo período. Ainda vale ressaltar que, na construção das soluções iniciais cada rota inicia-se em tempos específicos, esses tempos mínimos de saída do veículo  $v$  a partir do depósito são definidos na instância. Assim, a vizinhança ***addIdleTimeDepot(0, j)*** torna-se viável para avaliar os custos da rota  $v$  iniciando-se a partir de outros períodos.

#### 4.2.4 Mecanismo de Pertubação

Uma perturbação é selecionada aleatoriamente durante a execução. Logo, a perturbação aplicada possivelmente poderá piorar a solução, porém permitindo explorar diferentes espaços de busca.

- ***Multi-Swap(1,1)*** e ***Multi-Shift(1,0)***: Múltiplos movimentos ***Multi-Swap(1,1)*** e ***Multi-Shift(1,0)***. O número de movimentos executados é selecionado aleatoriamente no conjunto  $\{1, \dots, 4\}$ .
- ***Merge***: Duas rotas são concatenadas em uma nova rota.

### 4.3 Formulação *Set Covering*

A fase de intensificação do método híbrido proposto (Algoritmo 1 - linha 22) envolve a resolução exata de um problema de cobertura de conjuntos. Seja  $S \in ES$  uma solução (conjunto de rotas) da família de soluções elite  $ES$  (Algoritmo 1 - linha 12) e  $S_v \subseteq S$  sendo o subconjunto de rotas que contêm o cliente  $v \in N$  na solução  $S$ . Definimos a variável binária  $X_r$  associado a cada rota  $r \in S$  e  $C_r$  como o custo dessa rota. A formulação SC pode ser expressa da seguinte forma.

$$\text{Min } CO_2 = \sum_{S \in ES} \sum_{r \in S} C_r X_r \quad (4.21)$$

Sujeito a:

$$\sum_{S \in ES} \sum_{r \in S_v} X_r \geq 1, \quad \forall v \in N \quad (4.22)$$

$$X_r = X_s, \quad \forall S \in ES; \quad \forall r, s \in S, \text{ onde } r \text{ e } s \text{ compartilham o mesmo cliente} \quad (4.23)$$

$$X_r \in \{0, 1\}, \quad \forall r \in S; \quad \forall S \in ES \quad (4.24)$$

A função objetivo (4.21) encontra o conjunto de rotas de custo mínimo, dado que a restrição (4.22) garante que pelo menos uma rota passa por cada cliente. A viabilidade da solução é garantida pela restrição (4.23) que garante que as rotas com demandas fracionadas sempre estarão na mesma solução.

# Capítulo 5

## Resultados Computacionais

Este capítulo está dividido em duas seções principais. Na Seção 5.1 são apresentados os resultados para a abordagem exata do algoritmo. A Seção 5.2 descreve os resultados obtidos utilizando a meta-heurística pura (**MS-ILS**) além da versão híbrida do algoritmo de pós-processamento com a resolução do modelo de *Set Covering* (**MS-ILS-SC**). O algoritmo proposto foi desenvolvido na linguagem *C++* utilizando o compilador *g++* 4.8.2 e executado em um computador com *thread* única, processador *Intel®Core™i5-2310 CPU @ 2.90 GHz*, 8.0 GB de memória RAM e sob a plataforma de um Sistema Operacional *Linux (Ubuntu 14.04 LTS x64)*. Os modelos de programação linear inteira foram implementados utilizando o *framework* IBM ILOG CPLEX 12.6.1 e testados sob a mesma plataforma.

O banco de instâncias usado foi retirado do trabalho proposto por [66], as instâncias apresentam variados tamanhos e configurações. As instâncias utilizadas foram adaptadas pelos próprios autores a partir do banco de instâncias disponível no link *The Pollution Routing Problem Instance Library*<sup>1</sup>. O conjunto de instâncias incluem sete grupos de instâncias, cada grupo de instância contém vinte instâncias. As instâncias estão formatadas da seguinte maneira: **UK- $n \times m$ - $i$** , onde  $n$  é o número de clientes (10, 15, 25, ..., 100),  $m$  é o número de períodos, e  $i$  é o ID da instância (de 1 a 20).

A nomenclatura genérica para as tabelas apresentadas a partir deste capítulo é a seguinte: as colunas **Instância** apresentam as informações de cada instância, as duas próximas colunas **CO<sub>2</sub>** e **T** definem, respectivamente, o custo de carbono (medido em *kg/km*) e tempo de execução (medido em minutos); o resultado da meta-heurística MS-ILS é dado pela sigla **MS-ILS**, sendo este a média dos resultados obtidos para a execução

---

<sup>1</sup>PRPLIB: <http://www.apollo.management.soton.ac.uk/prplib.htm> - visualizado em 15.01.2018

de 5 casos de teste por instância; já os resultados para a versão híbrida do algoritmo é dado pela sigla **MS-ILS-SC**. Para os grupos de instância de  $10 \times 5, \dots, 25 \times 5$  o algoritmo **MS-ILS-SC** foi executado cinco vezes, para instâncias maiores o algoritmo **MS-ILS-SC** foi executado uma única vez, mesmo padrão utilizado na literatura.

Por praticidade, os resultados reportados em [66] são médias por grupo de instância, os autores criam uma média dos custos por grupo de instância tanto para abordagem exata quanto híbrida. O conjunto de instâncias  $10 \times 5$  é o único grupo de instância em que há detalhe dos custos em carbono e tempo de execução para cada instância e, apenas para o método exato tratado em [66]. Neste sentido, em algumas tabelas, como quesito de comparação apenas a média por grupo de instância será disposta nesse trabalho.

## 5.1 Abordagem Exata

As tabelas apresentadas nesta seção estão descritas como segue: a coluna **Gap (%)** representa o GAP, ou diferença relativa. Este GAP é definido como o intervalo entre a Melhor Solução Conhecida (**MSC**) da literatura e o melhor limite inferior encontrado pelo *solver* para cada formulação experimentada. Este limite inferior referenciado por **HGVRSP-Split-IP** e **HGVRSP-IP** representam respectivamente os resultados dos modelos MILP com e sem as restrições de entrega fracionada. O **Gap**<sup>1</sup> é dado por: **Gap**<sup>1</sup> =  $\frac{100 * (\text{HGVRSP-IP})}{(\text{MSC})} - 100$ , já o **Gap**<sup>2</sup> é dado por: **Gap**<sup>2</sup> =  $\frac{100 * (\text{HGVRSP-Split-IP})}{(\text{MSC})} - 100$ . A nomenclatura segue com valor **T(min)**, que representa o tempo total, em minutos, de execução do *solver*; **CO<sub>2</sub>** representa a média dos custos em carbono das soluções encontradas ao final da execução para o respectivo modelo aplicado.

O tempo limite definido para as execuções da abordagem exata foi definido em 7200 segundos, como os utilizados na literatura. As instâncias que tiveram a execução finalizada sob condições normais, i.e. tiveram o ótimo provado, o valor de GAP está destacado em negrito. Entretanto, algumas instâncias tiveram o tempo limite excedido, i.e., sem garantia de otimalidade, estas instâncias tiveram o valor de GAP destacado com  $\tau$ . Em outras instâncias a árvore de *branch* alcançou o limite físico de memória da máquina utilizada para teste, para essas instâncias, o valor do GAP está destacado com  $\star$ .

Avaliando a Tabela 5.1, os valores da primeira coluna são os resultados obtidos pela abordagem exata descrita em [66]. Embora exista alterações de algumas restrições, como demonstrado na Seção 4.1, as soluções reportadas em [66] são viáveis com o mesmo modelo reportado nesse trabalho. Neste sentido, comparando com o trabalho dos autores na sua

variante sem *Split*, o modelo aqui proposto consegue uma aproximação média de 0.01%, obtendo assim, uma média de 557.93 *kg/km* para o grupo de instâncias de tamanho dez. Examinando a linha de resultados para a instância **UK10x5-5** na versão sem entrega fracionada, o **HGVRSP-IP** obteve uma baixa taxa de precisão, cerca de 0.1% em relação ao ótimo. A instância **UK10x5-11** não foi avaliada uma vez que não foi possível encontrar soluções viáveis aplicando quaisquer abordagens.

MSC HGVRSP [66] <sup>a</sup>			HGVRSP-IP <sup>b</sup>			HGVRSP-Split-IP <sup>b</sup>		
Instâncias	CO <sub>2</sub>	T	CO <sub>2</sub>	T	Gap	CO <sub>2</sub>	T	Gap
UK-10x5-1	477.1	11.6	477.1	19.27	0.0%	471.44	31.64	-1.19%
UK-10x5-2	786.7	12.6	786.7	13.92	0.0%	752.34	24.73	-4.37%
UK-10x5-3	630.5	21.4	630.5	22.55	0.0%	601.28	32.73	-4.63%
UK-10x5-4	584.1	7.4	584.1	9.93	0.0%	581.51	26.51	-0.44%
UK-10x5-5	517.1	31.2	517.6	39.45	0.1%	511.31	28.37	-1.12%
UK-10x5-6	781.1	14.0	781.1	16.61	0.0%	778.55	41.13	-0.33%
UK-10x5-7	591.4	4.8	591.4	6.12	0.0%	582.33	26.28	-1.53%
UK-10x5-8	738.6	2.1	738.6	2.41	0.0%	733.06	37.42	-0.75%
UK-10x5-9	508.0	3.0	508.0	3.97	0.0%	502.33	41.72	-1.12%
UK-10x5-10	580.3	3.0	580.3	3.79	0.0%	579.47	22.26	-0.14%
UK-10x5-11	-	-	-	-	-	-	-	-
UK-10x5-12	512.3	2.8	512.3	3.13	0.0%	510.25	27.01	-0.40%
UK-10x5-13	588.6	6.2	588.6	8.74	0.0%	569.20	29.41	-3.30%
UK-10x5-14	540.4	11.0	540.4	12.59	0.0%	537.04	49.53	-0.62%
UK-10x5-15	364.6	47.1	364.6	59.31	0.0%	361.31	75.97	-0.90%
UK-10x5-16	540.3	44.2	540.3	51.32	0.0%	532.97	105.46	-1.36%
UK-10x5-17	534.5	2.5	534.5	3.95	0.0%	516.03	34.87	-3.46%
UK-10x5-18	446.2	11.4	446.2	13.25	0.0%	441.29	44.65	-1.10%
UK-10x5-19	443.3	0.4	443.3	1.24	0.0%	439.99	31.25	-0.75%
UK-10x5-20	434.8	7.5	434.8	8.33	0.0%	423.18	46.56	-2.67%
Média	557.88	12.85	557.93	15.78	0.01%	548.67	39.87	-1.59%

<sup>a</sup> Intel®Core™i5-2400S CPU @ 2.50 GHz

<sup>b</sup> Intel®Core™i5-2310 CPU @ 2.90 GHz

Tabela 5.1: Comparação entre soluções com e sem *Split* para o grupo de instância  $10 \times 5$ .

Ainda avaliando a Tabela 5.1, o custo para a instância **UK10x5-9** aplicando o modelo **HGVRSP-IP** chega a 508 *kg/km*, aplicando o modelo **HGVRSP-Split-IP** o custo da solução é reduzido para 502.33 *kg/km*, uma redução de -1.12% em relação a melhor solução conhecida. Os resultados obtidos pelo **HGVRSP-IP** foram semelhantes aos reportados em [66], perdendo apenas na instância **UK10x5-5**. Por outro lado, avaliando a média geral em tempo de CPU para o modelo **HGVRSP-IP** houve uma leve perda, mesmo com a média de geral de aproximação em torno de 0.01%, o modelo consegue ser, aproximadamente, 22% mais lento que a versão reportada em [66] para o grupo de instância  $10 \times 5$ .



Ao aplicar o modelo **HGVRSP-Split-IP** para a instância **UK10x5-3** temos que, o **HGVRSP-IP** gerou uma solução de  $630.5 \text{ kg/km}$ , o custo da solução é reduzido para  $601.28 \text{ kg/km}$  com a aplicação do modelo **HGVRSP-Split-IP**, um dos mais elevados ganhos na aproximação do ótimo, cerca de  $-4.63\%$ . Veja na Figura 5.1(a) que as soluções do modelo sem *Split* são levemente piores que a versão com *Split*. É fácil notar que a versão sem *Split* do problema representa um limitante superior para a solução ótima do problema com *Split* já que o problema HGVRSP-Split é um problema menos restrito que o HGVRSP. Por outro lado, o HGVRSP-Split possui uma combinatória muito mais explosiva que seu antecessor (devido ao maior grau de liberdade) que fica clara nos tempos de execução mais elevados que o HGVRSP. A Figura 5.1(b) ilustra esse comportamento com os tempos de execução da Melhor Solução Conhecida (MSC), **HGVRSP-IP** e **HGVRSP-Split-IP**.

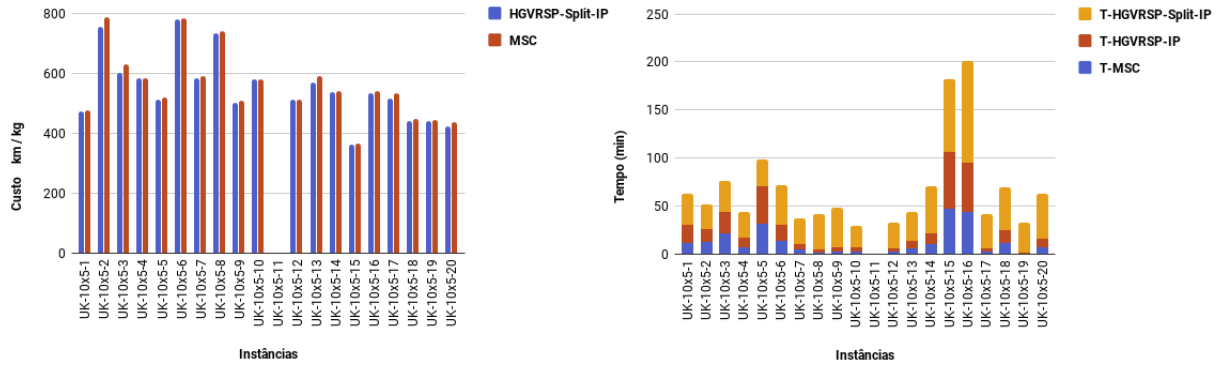
(a) Resultados para o  $Set 10 \times 5$ .(b) Tempos para o  $Set 10 \times 5$ .

Figura 5.1: Resultados para abordagem exata, instâncias com 10 vértices e 5 períodos.

O tempo computacional do modelo proposto variou significativamente de acordo com as instâncias, obtendo uma média geral de 15.78 minutos de CPU. Entretanto, o problema tornou-se inviável para instâncias maiores, uma vez que o tempo limite de 24 horas foi excedido na maioria dos casos, em contrapartida, instâncias que ficaram dentro do limite de 24 horas de CPU houve interrupção por falta de memória, o que acabou inviabilizando o modelo para instâncias maiores. Neste sentido, para instâncias maiores, é prático obter soluções quase ideais usando heurísticas ou uma abordagem híbrida.

O **HGVRSP-Split-IP** mostrou resultados competitivos em comparação com a literatura para instâncias de tamanho dez, embora o objetivo do estudo era propor uma heurística capaz de abordar instâncias maiores. Embora a qualidade da solução obtida pelo *solver* seja inferior à apresentada por [66], para instâncias de tamanho dez, o método exato ainda foi capaz de obter boas soluções, porém com um tempo superior, com um GAP médio de  $-1.59\%$  e tempo médio de 39.87 minutos. A próxima seção descreve os resultados obtidos pelas abordagens heurística e híbrida descritas ao longo das Seções 4.2 e 4.3.

## 5.2 Abordagem Heurística

Os parâmetros do algoritmo **MS-ILS** foram baseados no trabalho de [66], porém a quantidade de iterações sem melhora do algoritmo foi definida de forma empírica usando o pacote *irace* [40]. Assim, definiu-se os seguintes valores: tempo limite ( $maxT$ ) de 7200 segundos,  $maxIT = |N|$  e  $msIt = |N| + |M|$ . As heurísticas com e sem entrega fracionada são referenciadas nas tabelas desta seção respectivamente por **MS-ILS-Split** e **MS-ILS**.

O método **MS-ILS-Split** proposto nesse trabalho exige apenas alguns ajustes e a adição das estruturas descritas na Seção 4.2.3.5. Vale ressaltar que, para ambas as heurísticas aqui propostas, o mesmo método construtivo foi utilizado. Para a experimentação dos métodos, foi utilizado a biblioteca *Mersenne Twist*<sup>2</sup> de números aleatórios, considerou-se nos algoritmos, uma *seed* (semente) para cada uma das cinco execuções por instância.

Avaliando a Tabela 5.2, os valores da coluna **MSC** são os resultados obtidos pela abordagem exata em [66]. Neste sentido, comparando com o trabalho dos autores na sua variante sem *Split*, o **MS-ILS** aqui proposto consegue aproximar em média 2.97%, obtendo assim, uma média de 574.99 *kg/km* para o grupo de instâncias de tamanho dez. Analisando a linha de resultados para a instância **UK10x5-2**, o **MS-ILS** obteve uma das mais baixas taxas de precisão, cerca de 0.57% em relação ao ótimo.

MSC HGVRSP [66] <sup>a</sup>			MS-ILS			MS-ILS-Split			MS-ILS-SC		
Instâncias	CO <sub>2</sub>	T	CO <sub>2</sub>	T	Gap	CO <sub>2</sub>	T	Gap	CO <sub>2</sub>	T	Gap
UK-10x5-1	477.1	11.6	483.06	5.12	1.25%	472.72	18.23	-0.92%	472.28	25.11	-1.01%
UK-10x5-2	786.7	12.6	791.18	4.57	0.57%	762.81	11.45	-3.04%	752.41	16.72	-4.36%
UK-10x5-3	630.5	21.4	635.98	5.29	0.87%	603.45	12.67	-4.29%	602.89	22.87	-4.38%
UK-10x5-4	584.1	7.4	591.11	1.23	1.20%	582.52	14.49	-0.27%	582.43	17.32	-0.29%
UK-10x5-5	517.1	31.2	528.32	4.51	2.17%	511.56	18.58	-1.07%	511.54	21.06	-1.08%
UK-10x5-6	781.1	14.0	816.56	2.38	4.54%	781.11	29.45	0.00%	780.05	33.18	-0.13%
UK-10x5-7	591.4	4.8	597.78	1.29	1.08%	583.69	17.27	-1.30%	583.69	19.45	-1.30%
UK-10x5-8	738.6	2.1	811.49	1.45	9.87%	738.60	19.52	0.00%	735.22	24.17	-0.46%
UK-10x5-9	508	3.0	536.89	3.25	5.69%	506.21	21.28	-0.35%	502.89	25.94	-1.00%
UK-10x5-10	580.3	3.0	599.04	2.47	3.23%	580.30	12.24	0.00%	580.11	14.05	-0.03%
UK-10x5-11	-	-	-	-	-	-	-	-	-	-	-
UK-10x5-12	512.3	2.8	523.57	1.84	2.20%	512.31	15.31	0.00%	512.31	19.24	0.00%
UK-10x5-13	588.6	6.2	616.61	5.13	4.76%	571.29	19.11	-2.94%	569.87	21.09	-3.18%
UK-10x5-14	540.4	11.0	568.71	5.35	5.24%	540.42	36.54	0.00%	540.42	38.14	0.00%
UK-10x5-15	364.60	47.1	378.59	3.19	3.84%	364.60	55.22	0.00%	364.60	59.21	0.00%
UK-10x5-16	540.3	44.2	546.51	9.05	1.15%	537.31	74.14	-0.55%	535.18	78.82	-0.95%
UK-10x5-17	534.5	2.5	548.34	2.28	2.59%	521.06	23.53	-2.51%	519.01	25.29	-2.90%
UK-10x5-18	446.2	11.4	449.43	9.49	0.72%	443.48	32.25	-0.61%	443.48	35.12	-0.61%
UK-10x5-19	443.3	0.4	457.97	0.68	3.31%	443.31	19.38	0.00%	441.06	24.81	-0.51%
UK-10x5-20	434.8	7.5	443.84	6.15	2.08%	427.82	35.52	-1.61%	426.31	38.13	-1.95%
Média	557.88	12.85	574.99	3.93	2.97%	551.81	25.58	-1.02%	550.30	29.45	-1.27%

<sup>a</sup> Intel®Core™i5-2400S CPU @ 2.50 GHz

Tabela 5.2: Resultados para abordagem heurística, grupo de instância 10 × 5.

<sup>2</sup> Mersenne Twist v. 1.5: <https://www.cs.hmc.edu/~geoff/mtwist.html>, visualizado em 05.01.2018

Ainda avaliando a Tabela 5.2, a heurística **MS-ILS-Split** consegue boas reduções de custo para o grupo de instâncias de tamanho dez, temos na instância **UK10x5-9** uma redução no custo de 536.89 *kg/km* para 506.21 *kg/km*, uma redução de 5.69% em relação ao **MS-ILS** e uma redução de -0.35% do ótimo. Além disso, aplicando a abordagem híbrida **MS-ILS-SC** na sua variante com *Split* para a instância **UK10x5-9**, temos uma solução de custo 502.89 *kg/km*, uma redução de -1% em relação ao ótimo. Veja na Figura 5.2(a) que as soluções do **MS-ILS** são levemente piores em custo do que a heurística **MS-ILS-Split**, que por sua vez é inferior em relação aos resultados de **MS-ILS-SC**. Como visto nos resultados dos métodos exatos, o tempo de execução para os métodos com *Split* são bem superiores ao método sem *Split* (Figura 5.2(b)) devido a sua maior combinatória no espaço de soluções.

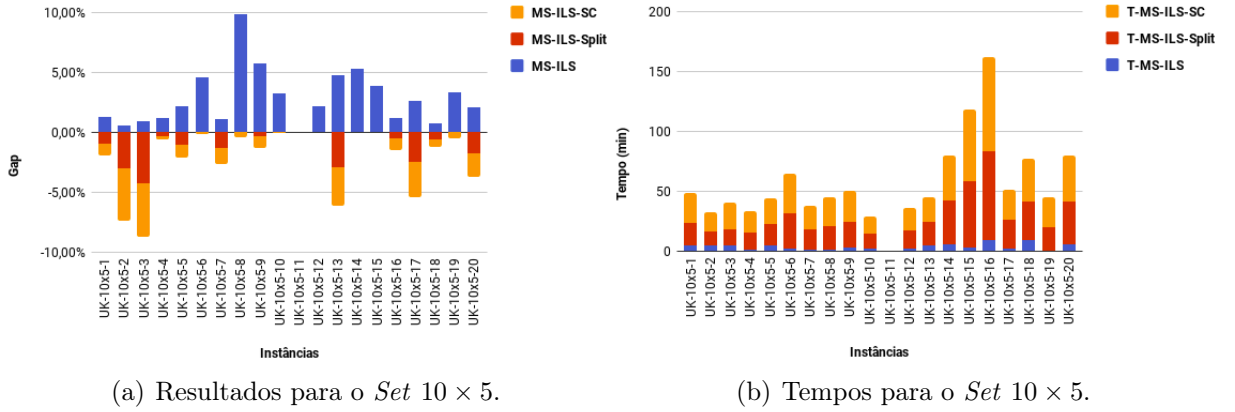


Figura 5.2: Resultados para abordagem heurística, instâncias com 10 vértices e 5 períodos.

Devido a inviabilidade de solucionar instâncias de grande porte, o método exato só foi avaliado para instâncias de tamanho dez. Neste sentido, a Tabela 5.3 faz uma breve descrição das médias obtidas para os sete grupos de instâncias, cada linha da Tabela 5.3 representa a média dos custos para cada grupo de instância avaliando apenas o melhor método heurístico reportado em [66], o problema é abordado pelos autores com o uso de três heurísticas distintas, sendo elas: **MSC<sub>1</sub>** (*Nearest-Neighbor*), **MSC<sub>2</sub>** (*Savings heuristic*) e **MSC<sub>3</sub>** (*Insertion heuristic*). Neste sentido, a comparação é realizada apenas com a heurística de melhor custo em carbono para cada instância.

A meta-heurística **MS-ILS** ganha em comparação com todas as heurísticas usadas no artigo, isso ocorre tanto na versão com e sem *Split*. Ainda sobre os resultados reportados na Tabela 5.3, a heurística **MS-ILS** reduziu, no geral, os custos em -1.83% na sua versão sem *Split* e -4.75% com *Split*, demonstrando assim, que houve bons resultados na média geral. Ao avaliar os tempos de CPU, ambos os métodos aqui propostos perdem quando comparados com a heurística reportada em [66].

MSC HGVRSP [66] <sup>a</sup>			MS-ILS			MS-ILS-Split		
Grupo de Inst.	CO <sub>2</sub>	T	CO <sub>2</sub>	T	Gap	CO <sub>2</sub>	T	Gap
10 × 5	635.9	1.0	574.99	3.93	<b>-9.58%</b>	551.81	25.58	<b>-13.22%</b>
15 × 5	719.9	1.0	728.05	5.31	<b>1.13%</b>	705.01	27.17	<b>-2.07%</b>
20 × 5	1084.2	1.0	1078.13	8.14	<b>-0.56%</b>	1033.97	29.39	<b>-4.63%</b>
25 × 5	1066.3	1.0	1054.22	10.62	<b>-1.13%</b>	1017.57	35.28	<b>-4.57%</b>
50 × 5	2080.3	2.8	2066.04	18.23	<b>-0.69%</b>	2024.31	41.91	<b>-2.69%</b>
75 × 5	2977.4	7.0	2932.31	22.41	<b>-1.51%</b>	2878.06	49.23	<b>-3.34%</b>
100 × 5	3766.3	13.0	3748.52	28.05	<b>-0.47%</b>	3662.84	60.89	<b>-2.75%</b>
Média		3.82	1740.32	13.81	<b>-1.83%</b>	1696.22	38.49	<b>-4.75%</b>

<sup>a</sup> Intel®Core™i5-2400S CPU @ 2.50 GHz

Tabela 5.3: Resultados para os sete grupos de instâncias utilizando a heurística.

A Tabela 5.4 faz uma breve descrição das médias obtidas para os sete grupos de instâncias utilizando o **MS-ILS-SC**. Cada linha da tabela representa a média dos custos para cada grupo de instância avaliando apenas o método híbrido reportado em [66]. Os autores utilizam três heurísticas distintas para encontrar uma solução inicial e fixam parte dessa solução no modelo. O método híbrido reportado em [66] fixa as estruturas das rotas no modelo e usa o *solver* para montar as agendas dos veículos.

HGVRSP-hybrid [66] <sup>a</sup>			MS-ILS-SC <sup>b</sup>		
Grupo de Inst.	CO <sub>2</sub>	T	CO <sub>2</sub>	T	Gap
10 × 5	557.9	7.73	550.34	29.45	<b>-1.36%</b>
15 × 5	746.1	12.91	697.26	31.26	<b>-6.55%</b>
20 × 5	958.7	17.98	946.98	33.02	<b>-1.22%</b>
25 × 5	936.0	31.06	928.17	39.91	<b>-0.84%</b>
50 × 5	1710.2	59.00	1697.88	45.29	<b>-0.72%</b>
75 × 5	2483.5	89.00	2463.39	53.52	<b>-0.81%</b>
100 × 5	3149.8	98.00	3128.17	68.14	<b>-0.69%</b>
Média	1506.02	45.09	1487.45	42.94	<b>-1.74%</b>

<sup>a</sup> Intel®Core™i5-2400S CPU @ 2.50 GHz

<sup>b</sup> Intel®Core™i5-2310 CPU @ 2.90 GHz

Tabela 5.4: Comparação dos resultados para o método híbrido.

As soluções obtidas pelo **MS-ILS-SC** apresentam uma relativa melhora em relação aos resultados reportados em [66]. O **MS-ILS-SC** consegue gerar um *gap* na média de  $-1.74\%$  em relação a todos os grupos de instância, em contrapartida o modelo híbrido perde em tempo computacional para instancias menores. Além disso, o método consegue ganhos significativos em tempo apenas para os grupos de instância  $50 \times 5$ ,  $75 \times 5$  e  $100 \times 5$ . Ao realizar uma avaliação geral do **MS-ILS-SC**, o método consegue reduzir, no geral, o tempo de CPU em  $4,81\%$  quando comparado com os tempos do estado-da-arte.

# Capítulo 6

## Conclusões e Trabalhos Futuros

Esta dissertação apresentou uma nova abordagem para o HGVRSP, usando uma metodologia baseada na meta-heurística *Iterated Local Search* com uma busca local intensiva através de várias vizinhanças clássicas para problemas de roteamento. Os testes realizados provaram que a metodologia proposta foi capaz de encontrar soluções de boa qualidade, em muitos casos reduzindo o custo. Avaliando a heurística com *Split*, das 20 instâncias de tamanho 10 testadas, 13 apresentaram melhora e 7 empataram, um ganho em 75% das instâncias. Mesmo o **MS-ILS** obtendo gaps na média de 2.96% em relação a melhor solução conhecida, o mesmo torna-se atrativo uma vez que as suas sucessivas iterações tendem a convergir para bons resultados com um baixo tempo computacional.

Como estudo inicial, o algoritmo provou ser promissor na exploração dessa variante do HGVRSP, entretanto mais estudos são necessários. Além disso, existe a possibilidade de estender essa metodologia para outras variantes do HGVRSP, por exemplo, o HGVRSP com restrições de veículos que podem atender certos clientes, que são os de maior aplicação prática. Outra possibilidade seria definir a velocidade como uma variável de decisão para avaliar a melhor velocidade dentro de um arco, além de uma versão sem paradas nos arcos. Em outro cenário, pode-se ponderar a quantidade de gasolina como variável de decisão, neste sentido, pode-se avaliar a quantidade mínima de gasolina para executar um trajeto de forma a minimizar tanto os custos de carbono quanto os custos operacionais.

Para abordagem exata, deve-se realizar um estudo mais amplo de modo a propor novos modelos de PLI que tragam uma maior aproximação da envoltória convexa do problema. Outra meta é apresentar novas inequações válidas e novos cortes que possivelmente venham a ser facetas e que devem ser capazes de garantir uma melhoria nos resultados reportados. Além disso, pretende-se obter limites inferiores melhores através de uma técnica de geração de colunas que pode garantir a prova do ótimo de outras instâncias, uma vez que existem bons limites superiores obtidos pela heurística proposta.

Um mecanismo de filtragem probabilístico pode ser implementado no passo *Multi-Start* do algoritmo de modo a evitar iterações no ILS nos casos em que a solução construída possivelmente não leve a uma boa solução final. Por fim, o uso de Estruturas de Dados Auxiliares (EDAs) mais eficientes também pode ser levado em consideração de modo a evitar o cálculo de vizinhanças que não são interessantes em algum ponto da busca local. Estas poucas implementações devem levar a um ganho significativo de desempenho computacional não só na abordagem heurística mas também na híbrida.

Como proposta de trabalhos futuros, pretende-se trabalhar com técnicas construtivas mais eficientes, técnicas capazes de gerar boa variabilidade nas soluções sem perda de qualidade. Desta forma, é interessante realizar um estudo mais amplo do impacto dos parâmetros do algoritmo e das vizinhanças, podendo inclusive realizar adição de novas estruturas ou até mesmo a remoção de estruturas não muito eficientes para o problema. Outra proposta, é a aplicação de variações híbridas mais eficientes do algoritmo como recurso de pré-otimização em métodos exatos.

# Referências

- [1] ARCHETTI, C., BIANCHESSI, N., SPERANZA, M. G. A column generation approach for the split delivery vehicle routing problem. *Networks* 58, 4 (2011), 241–254.
- [2] ARCHETTI, C., SPERANZA, M. G. The split delivery vehicle routing problem: A survey. In *The vehicle routing problem: Latest advances and new challenges*. Springer, 2008, p. 103–122.
- [3] BARTH, M., BORIBOONSOMSIN, K. Real-world carbon dioxide impacts of traffic congestion. *Transportation Research Record: Journal of the Transportation Research Board* 2058, 2058 (2008), 163–171.
- [4] BARTH, M., YOUNGLOVE, T., SCORA, G. Development of a heavy-duty diesel modal emissions and fuel consumption model. *California Partners for Advanced Transit and Highways (PATH)* (2005).
- [5] BEKTAŞ, T., LAPORTE, G. The pollution-routing problem. *Transportation Research Part B: Methodological* 45, 8 (2011), 1232–1250.
- [6] BODIN, L., GOLDEN, B. Classification in vehicle routing and scheduling. *Networks* 11, 2 (1981), 97–108.
- [7] BOUDIA, M., PRINS, C., REGHIOUI, M. An effective memetic algorithm with population management for the split delivery vehicle routing problem. In *International Workshop on Hybrid Metaheuristics* (2007), Springer, p. 16–30.
- [8] CAMPOS, V., CORBERÁN, A., MOTA, E. A scatter search algorithm for the split delivery vehicle routing problem. *Advances in computational intelligence in transport, logistics, and supply chain management* (2008), 137–152.
- [9] CHEN, H.-K., HSUEH, C.-F., CHANG, M.-S. The real-time time-dependent vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review* 42, 5 (2006), 383–408.
- [10] CHEN, S., GOLDEN, B., WASIL, E. The split delivery vehicle routing problem: Applications, algorithms, test problems, and computational results. *Networks* 49, 4 (2007), 318–329.
- [11] CLARKE, G., WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research* 12, 4 (1964), 568–581.
- [12] COELHO, B. N., COELHO, V. N., COELHO, I. M., OCHI, L. S., K., R. H., ZUIDEMA, D., LIMA, M. S., DA COSTA, A. R. A multi-objective green uav routing problem. *Computers & Operations Research* 88, Supplement C (2017), 306–315.

- [13] COELHO, V. N., COELHO, I. M., COELHO, B. N., DE OLIVEIRA, G. C., BARBOSA, A. C., PEREIRA, L., DE FREITAS, A., SANTOS, H. G., OCHI, L. S., GUIMARÃES, F. G. A communitarian microgrid storage planning system inside the scope of a smart city. *Applied Energy* 201, Supplement C (2017), 371–381.
- [14] COHEN, N. *Green cities: An A-to-Z guide*, vol. 4. Sage, 2011.
- [15] DANTZIG, G. B., RAMSER, J. H. The truck dispatching problem. *Management science* 6, 1 (1959), 80–91.
- [16] DEMIR, E., BEKTAŞ, T., LAPORTE, G. An adaptive large neighborhood search heuristic for the pollution-routing problem. *European Journal of Operational Research* 223, 2 (2012), 346 – 359.
- [17] DROR, M., TRUDEAU, P. Savings by split delivery routing. *Transportation Science* 23, 2 (1989), 141–145.
- [18] EKŞIOĞLU, B., VURAL, A. V., REISMAN, A. The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering* 57, 4 (2009), 1472–1483.
- [19] FAGERHOLT, K., LAPORTE, G., NORSTAD, I. Reducing fuel emissions by optimizing speed on shipping routes. *Journal of the Operational Research Society* 61, 3 (2010), 523–529.
- [20] FIGLIOZZI, M. Vehicle routing problem for emissions minimization. *Transportation Research Record: Journal of the Transportation Research Board* 2197, 2197 (2010), 1–7.
- [21] FRANCESCHETTI, A., HONHON, D., VAN WOENSEL, T., BEKTAŞ, T., LAPORTE, G. The time-dependent pollution-routing problem. *Transportation Research Part B: Methodological* 56 (2013), 265–293.
- [22] GLOVER, F., LAGUNA, M., MARTI, R. *Scatter Search and Path Relinking: Advances and Applications*. Springer US, Boston, MA, 2003, p. 1–35.
- [23] GOLDEN, B. L., RAGHAVAN, S., WASIL, E. A. *The vehicle routing problem: latest advances and new challenges*, vol. 43. Springer Science & Business Media, 2008.
- [24] HANSEN, P., MLADENović, N., PÉREZ, J. A. M. Variable neighbourhood search: methods and applications. *Annals of Operations Research* 175, 1 (2010), 367–407.
- [25] HOLLAND, J. H. Adaptation in natural and artificial systems. an introductory analysis with application to biology, control, and artificial intelligence. *Ann Arbor, MI: University of Michigan Press* (1975).
- [26] HVATTUM, L. M., NORSTAD, I., FAGERHOLT, K., LAPORTE, G. Analysis of an exact algorithm for the vessel speed optimization problem. *Networks* 62, 2 (2013), 132–135.
- [27] IEA, O. Emissions from fossil fuel combustion: Highlights. *International Energy Agency* (2017).



- [28] IRNICH, S. Resource extension functions: Properties, inversion, and generalization to segments. *OR Spectrum* 30, 1 (2008), 113–148.
- [29] JABALI, O., WOENSEL, T., DE KOK, A. Analysis of travel times and co2 emissions in time-dependent vehicle routing. *Production and Operations Management* 21, 6 (2012), 1060–1074.
- [30] KARA, I., KARA, B., YETIS, M. Energy minimizing vehicle routing problem. *Combinatorial optimization and applications* (2007), 62–71.
- [31] KINDERVATER, G. A., SAVELSBERGH, M. W. Vehicle routing: handling edge exchanges. *Local search in combinatorial optimization* (1997), 337–360.
- [32] KIRKPATRICK, S., GELATT, C. D., VECCHI, M. P., OTHERS. Optimization by simulated annealing. *science* 220, 4598 (1983), 671–680.
- [33] KRAMER, R., SUBRAMANIAN, A., VIDAL, T., LUCÍDIO DOS ANJOS, F. C. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research* 243, 2 (2015), 523–539.
- [34] KUO, Y. Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers & Industrial Engineering* 59, 1 (2010), 157–165.
- [35] KUO, Y., WANG, C.-C. Optimizing the vrp by minimizing fuel consumption. *Management of Environmental Quality: An International Journal* 22, 4 (2011), 440–450.
- [36] LAI, K.-H., WONG, C. W. Green logistics management and performance: Some empirical evidence from chinese manufacturing exporters. *Omega* 40, 3 (2012), 267–282.
- [37] LAPORTE, G., GENDREAU, M., POTVIN, J.-Y., SEMET, F. Classical and modern heuristics for the vehicle routing problem. *International transactions in operational research* 7, 4-5 (2000), 285–300.
- [38] LENSTRA, J. K., KAN, A. Complexity of vehicle routing and scheduling problems. *Networks* 11, 2 (1981), 221–227. LENSTRA, J. K. and RINNOOY KAN, A. H. G.
- [39] LIN, C., CHOY, K. L., HO, G. T. S., CHUNG, S. H., LAM, H. Y. Survey of Green Vehicle Routing Problem: Past and future trends. *Expert Systems with Applications* 41, 4 PART 1 (2014), 1118–1138.
- [40] LÓPEZ-IBÁÑEZ, M., DUBOIS-LACOSTE, J., STÜTZLE, T., BIRATTARI, M. The irace package, iterated race for automatic algorithm configuration. Relatório Técnico, Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles, Belgium, 2011.
- [41] LOURENÇO, H. R., MARTIN, O. C., STÜTZLE, T. Iterated local search: Framework and applications. In *Handbook of Metaheuristics*. Springer, 2010, p. 363–397.
- [42] MADEN, W., EGGLESE, R., BLACK, D. Vehicle routing and scheduling with time-varying data: A case study. *Journal of the Operational Research Society* 61, 3 (2010), 515–522.

- [43] MATOS, M. R. S., DE MENEZES FROTA, Y. A., OCHI, L. S. Problema de roteamento de veículos voltado a minimização de emissões de dióxido de carbono. *XLIX SBPO. Brazil* (2017), 3808–3819.
- [44] MORENO, L., DE ARAGÃO, M. P., UCHOA, E. Improved lower bounds for the split delivery vehicle routing problem. *Operations Research Letters* 38, 4 (2010), 302–306.
- [45] NAGATA, Y., BRÄYSY, O., DULLAERT, W. A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. *Computers & operations research* 37, 4 (2010), 724–737.
- [46] NORSTAD, I., FAGERHOLT, K., LAPORTE, G. Tramp ship routing and scheduling with speed optimization. *Transportation Research Part C: Emerging Technologies* 19, 5 (2011), 853–865.
- [47] PENG, Y., WANG, X. Research on a vehicle routing schedule to reduce fuel consumption. In *Measuring Technology and Mechatronics Automation, 2009. ICMTMA'09. International Conference on* (2009), vol. 3, IEEE, p. 825–827.
- [48] PENNA, P. *Um algoritmo unificado para uma classe de problemas de roteamento de veículos com frota heterogênea*. Tese de Doutorado, PhD thesis, Universidade Federal Fluminense - UFF, 2013.
- [49] PENNA, P. H. V., SUBRAMANIAN, A., OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics* (2013), 1–32.
- [50] SABERI, M., VERBAS, İ. Ö. Continuous approximation model for the vehicle routing problem for emissions minimization at the strategic level. *Journal of Transportation Engineering* 138, 11 (2012), 1368–1376.
- [51] SCOTT, C., URQUHART, N., HART, E. Influence of topology and payload on co 2 optimised vehicle routing. *Applications of Evolutionary Computation* (2010), 141–150.
- [52] SILVA, M. M., SUBRAMANIAN, A., OCHI, L. S. An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research* 53 (2015), 234–249.
- [53] SOLOMON, M. M. Vehicle routing and scheduling with time window constraints: Models and algorithms. Relatório Técnico 01451798, University of Pennsylvania, 1984.
- [54] SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research* 35, 2 (1987), 254–265.
- [55] SOUZA, M. J., COELHO, I. M., RIBAS, S., SANTOS, H. G., MERSCHMANN, L. H. D. C. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research* 207, 2 (2010), 1041–1051.
- [56] SOYSAL, M., BLOEMHOF-RUWAARD, J. M., BEKTAŞ, T. The time-dependent two-echelon capacitated vehicle routing problem with environmental considerations. *International Journal of Production Economics* 164 (2015), 366–378.

- [57] SUBRAMANIAN, A., DRUMMOND, L. M. D. A., BENTES, C., OCHI, L. S., FARIAS, R. A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research* 37, 11 (2010), 1899–1911.
- [58] SUBRAMANIAN, A., PENNA, P. H. V., UCHOA, E., OCHI, L. S. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research* 221, 2 (2012), 285–295.
- [59] TOTH, P., VIGO, D. Chapter 1: An overview of vehicle routing problems. *The Vehicle Routing Problem. Society for Industrial and Applied Mathematics Philadelphia, PA, USA* (2002).
- [60] UBEDA, S., ARCELUS, F., FAULIN, J. Green logistics at eroski: A case study. *International Journal of Production Economics* 131, 1 (2011), 44–51.
- [61] VIDAL, T., CRAINIC, T. G., GENDREAU, M., PRINS, C. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research* 231, 1 (2013), 1–21.
- [62] VIDAL, T., CRAINIC, T. G., GENDREAU, M., PRINS, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research* 40, 1 (2013), 475–489.
- [63] WOODCOCK, J., EDWARDS, P., TONNE, C., ARMSTRONG, B. G., ASHIRU, O., BANISTER, D., BEEVERS, S., CHALABI, Z., CHOWDHURY, Z., COHEN, A., FRANCO, O. H., HAINES, A., HICKMAN, R., LINDSAY, G., MITTAL, I., MOHAN, D., TIWARI, G., WOODWARD, A., ROBERTS, I. Public health benefits of strategies to reduce greenhouse-gas emissions: urban land transport. *The Lancet* 374, 9705 (2009), 1930 – 1943.
- [64] XIAO, Y., KONAK, A. Green vehicle routing problem with time-varying traffic congestion. In *Proceedings of the 14th INFORMS Computing Society Conference* (2015), p. 134–148.
- [65] XIAO, Y., KONAK, A. A simulating annealing algorithm to solve the green vehicle routing & scheduling problem with hierarchical objectives and weighted tardiness. *Applied Soft Computing* 34 (2015), 372 – 388.
- [66] XIAO, Y., KONAK, A. The heterogeneous green vehicle routing and scheduling problem with time-varying traffic congestion. *Transportation Research Part E: Logistics and Transportation Review* 88 (2016), 146–166.
- [67] XIAO, Y., ZHAO, Q., KAKU, I., XU, Y. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research* 39, 7 (2012), 1419–1431.