

**ALGORITMOS PARA O PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS CAPACITADO  
COM RESTRIÇÕES DE CARREGAMENTO  
BIDIMENSIONAL**



VITOR ANDRADE ALMEIDA DE SOUZA

**ALGORITMOS PARA O PROBLEMA DE  
ROTEAMENTO DE VEÍCULOS CAPACITADO  
COM RESTRIÇÕES DE CARREGAMENTO  
BIDIMENSIONAL**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais - Departamento de Ciência da Computação como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: GERALDO ROBSON MATEUS

Belo Horizonte

Março de 2013

© 2013, Vitor Andrade Almeida de Souza.  
Todos os direitos reservados.

Souza, Vitor Andrade Almeida de

S729a      Algoritmos para o Problema de Roteamento de  
Veículos Capacitado com Restrições de Carregamento  
Bidimensional / Vitor Andrade Almeida de Souza. —  
Belo Horizonte, 2013  
xxiv, 110 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais - Departamento de Ciência da  
Computação

Orientador: Geraldo Robson Mateus

1. Computação - Teses. 2. Otimização Combinatória  
- Teses. 3. Programação heurística - Teses

I. Orientador. II. Título

CDU 519.6\*62.(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

Algoritmos para o problema de roteamento de veículos capacitado com  
restrições de carregamento bidimensional

**VITOR ANDRADE ALMEIDA DE SOUZA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

A handwritten signature in blue ink, appearing to read "G. Mateus".

PROF. GERALDO ROBSON MATEUS - Orientador  
Departamento de Ciência da Computação - UFMG

A handwritten signature in blue ink, appearing to read "André Gustavo dos Santos".

PROF. ANDRÉ GUSTAVO DOS SANTOS  
Departamento de Informática - UFV

A handwritten signature in blue ink, appearing to read "Luciana Salete Buriol".

PROFA. LUCIANA SALETE BURIOL  
Instituto de Informática - UFRGS

A handwritten signature in blue ink, appearing to read "Sebastián Alberto Urrutia".

PROF. SEBASTIÁN ALBERTO URRUTIA  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 18 de março de 2013.



*Aos meus avós,  
José (in memoriam)  
e Laudelina.*





# Agradecimentos

Ao meu orientador, professor Geraldo Robson Mateus, agradeço pela confiança, paciência, dedicação e ensinamentos.

Ao meu tio Elton, avó Laudelina, mãe e irmãos, agradeço por terem sempre me ajudado em tudo, mesmo à distância.

A Nathee, minha namorada, agradeço por estar ao meu lado tanto nos momentos bons quanto nos ruins e por ter suportado esses anos de distância. Você faz parte dessa vitória.

Ao professor Thiago, Ramon e Vinícius, agradeço a oportunidade de termos trabalhado juntos no Google ROADEF/EURO Challenge 2012, o qual foi muito importante para o meu aprendizado.

Aos amigos do LaPO, em especial Ramon e Vinícius, agradeço aos bons momentos vivenciados juntos e às ajudas nos momentos de dificuldade.

Agradeço àqueles que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

Agradeço também à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), que financiou o meu mestrado.



# Resumo

Esta dissertação aborda o Problema de Roteamento de Veículos Capacitado com Restrições de Carregamento Bidimensional. Dado um depósito central e um conjunto de clientes, onde cada um demanda uma quantidade específica de itens, o problema consiste em definir rotas com distâncias percorridas mínimas para uma frota homogênea de veículos que realiza o atendimento aos clientes. Os itens possuem formas retangulares, devem ser transportados de modo que não haja sobreposição e impedimento entre eles e, em alguns casos, restrições de carregamento sequencial, relacionados à ordem de visita dos clientes, são exigidas. Para a resolução do problema, foram propostas duas abordagens híbridas de heurísticas e Geração de Colunas. Além disso, um algoritmo *Branch-and-Cut* da literatura foi usado para resolver uma reformulação do modelo original do problema. Os métodos desenvolvidos foram avaliados por meio das instâncias usadas na literatura e os resultados foram comparados aos previamente publicados. Os métodos híbridos obtiveram resultados satisfatórios, algumas vezes iguais aos ótimos conhecidos, e o *Branch-and-Cut* conseguiu atestar a otimalidade para várias instâncias.

**Palavras-chave:** Roteamento de Veículos, Carregamento Bidimensional, Geração de Colunas, Metaheurística, Otimização Combinatória.



# Abstract

This work addresses the Capacitated Vehicle Routing Problem with two-dimensional loading constraints. Given a central depot and a set of clients, where each demands a specific amount of items, the problem aims to define minimum cost routes for a fleet of homogeneous vehicles that performs customer service. The items have rectangular shapes, they must be transported in a way that there is no overlap between them, and in some cases, sequential loading restrictions, related to the order of visiting the customers, are required. To solve the problem, two hybrid approaches combining heuristics and Column Generation are proposed. Furthermore, the literature's Branch-and-Cut was used to solve a reformulation of the original model of the problem. The methods developed were evaluated by means of the instances used in the literature and the results were compared with those previously published. The hybrid methods achieve satisfactory results, sometimes equal to the optima known, and the Branch-and-Cut could attest to optimality for several instances.

**Keywords:** Vehicle Routing, Two-dimensional Loading, Column Generation, Matheuristic, Combinatorial Optimization.



# Lista de Figuras

1.1	Carregamento irrestrito e sequencial. . . . .	3
1.2	Rota inviável e rota viável para o <b>2L-CVRP</b> sequencial, considerando os mesmos clientes. . . . .	5
4.1	Inserindo um item no compartimento do veículo. . . . .	20
4.2	Cálculo do perímetro tocante para as heurísticas (a) <i>Max Touching Perimeter</i> e (b) <i>Max Touching Perimeter No Walls</i> . . . . .	22
4.3	Cálculo das superfícies retangulares de cada posição disponível na <i>posList</i> . . . . .	23
4.4	Carregamento parcial com a linha de contorno e os <i>corner points</i> destacados. . . . .	27
4.5	Sucessão de inserções na árvore de viabilidade. . . . .	30
5.1	Exemplo de um movimento da busca local <i>Inversion</i> . . . . .	37
5.2	Exemplo de um movimento da busca local <i>Exchange</i> . . . . .	37
5.3	Exemplo de um movimento da busca local <i>1-OPT</i> . . . . .	38
5.4	Exemplo de um movimento da busca local <i>2-OPT</i> . . . . .	38





# Lista de Tabelas

2.1	Características adotadas pelos trabalhos da literatura na resolução do 2L-CVRP	12
6.1	Características da geração dos itens. . . . .	54
6.2	Características das instâncias. . . . .	55
6.3	Comparação das instâncias da Classe 1 com as <i>Características <math>T_i</math></i> . . . . .	58
6.4	Comparação das instâncias das Classes 2-5 com as <i>Características <math>T_i</math></i> no caso sequencial. . . . .	58
6.5	Comparação das instâncias das Classes 2-5 com as <i>Características <math>T_i</math></i> no caso irrestrito. . . . .	60
6.6	Comparação das instâncias da Classe 1 com as <i>Características <math>Tr</math></i> . . . . .	62
6.7	Comparação das instâncias das Classes 2-5 com as <i>Características <math>Tr</math></i> no caso sequencial. . . . .	62
6.8	Comparação das instâncias das Classes 2-5 com as <i>Características <math>Tr</math></i> no caso irrestrito. . . . .	64
A.1	Resultados detalhados do CGV/CGN para a Classe 1 com as <i>Características <math>T_i</math></i> . . . . .	80
A.2	Resultados detalhados do CGV para as Classes 2-5 com as <i>Características <math>T_i</math></i> no caso sequencial. . . . .	81
A.3	Resultados detalhados do CGV para as Classes 2-5 com as <i>Características <math>T_i</math></i> no caso irrestrito. . . . .	83
A.4	Resultados detalhados do CGN para as Classes 2-5 com as <i>Características <math>T_i</math></i> no caso sequencial. . . . .	86
A.5	Resultados detalhados do CGN para as Classes 2-5 com as <i>Características <math>T_i</math></i> no caso irrestrito. . . . .	89
A.6	Resultados detalhados do CGV/CGN para a Classe 1 com as <i>Características <math>Tr</math></i> . . . . .	93
A.7	Resultados detalhados do CGV para as Classes 2-5 com as <i>Características <math>Tr</math></i> no caso sequencial. . . . .	94

A.8	Resultados detalhados do <b>CGV</b> para as Classes 2-5 com as <i>Características Tr</i> no caso irrestrito. . . . .	96
A.9	Resultados detalhados do <b>CGN</b> para as Classes 2-5 com as <i>Características Tr</i> no caso sequencial. . . . .	99
A.10	Resultados detalhados do <b>CGN</b> para as Classes 2-5 com as <i>Características Tr</i> no caso irrestrito. . . . .	102
A.11	Resultados detalhados do <b>BC</b> para a Classe 1 com as <i>Características Tr</i> . . .	105
A.12	Resultados detalhados do <b>BC</b> para as Classes 2-5 com as <i>Características Tr</i> no caso sequencial. . . . .	106
A.13	Resultados detalhados do <b>BC</b> para as Classes 2-5 com as <i>Características Tr</i> no caso irrestrito. . . . .	108

# Lista de Algoritmos

1	Coleção de heurísticas para o carregamento. . . . .	23
2	Método <i>Geração de Colunas</i> . . . . .	35
3	Heurística <i>ILS-1</i> . . . . .	39
4	Heurística <i>ILS-2</i> . . . . .	40
5	Método <i>Gera Pool Inicial de Colunas</i> . . . . .	41
6	Método <i>Gera Novas Colunas</i> . . . . .	46
7	Heurística <i>CGV</i> . . . . .	47
8	Heurística <i>CGN</i> . . . . .	49
9	Método <i>Verifica Carregamento BC</i> . . . . .	51



# Lista de Siglas

<b>1BPP</b>	<i>One-dimensional Bin Packing Problem</i>
<b>1CSP</b>	<i>One-dimensional Cutting Stock Problem</i>
<b>2BPP</b>	<i>Two-dimensional Bin Packing Problem</i>
<b>2CPP</b>	<i>Two-dimensional Cutting and Packing Problems</i>
<b>2CSP</b>	<i>Two-dimensional Cutting Stock Problem</i>
<b>2D-KP</b>	<i>Two-dimensional Knapsack Problem</i>
<b>2L-CVRP</b>	<i>Capacitated Vehicle Routing Problem with Two-dimensional Loading Constraints</i>
<b>2OPP</b>	<i>Two-dimensional Orthogonal Packing Problem</i>
<b>2SPP</b>	<i>Two-dimensional Strip Packing Problem</i>
<b>3L-CVRP</b>	<i>Capacitated Vehicle Routing Problem with Three-dimensional Loading Constraints</i>
<b>3L-VRPTW</b>	<i>Vehicle Routing Problem with Time Windows and Three-dimensional Loading Constraints</i>
<b>BB</b>	<i>Branch-and-Bound</i>
<b>BC</b>	<i>Branch-and-Cut</i>
<b>BCP</b>	<i>Branch-and-Cut-and-Price</i>
<b>BP</b>	<i>Branch-and-Price</i>
<b>CCCs</b>	<i>capacity-cut constraints</i>
<b>CG</b>	<i>Column Generation</i>
<b>CGN</b>	<i>Column Generation without Loading Verification in Pricing</i>

<b>CGV</b>	<i>Column Generation with Loading Verification in Pricing</i>
<b>CVRP</b>	<i>Capacitated Vehicle Routing Problem</i>
<b>ESPPRC</b>	<i>Elementary Shortest Path Problem with Resource Constraints</i>
<b>GRASP</b>	<i>Greedy Randomized Adaptative Search Procedure</i>
<b>GRASP x ELS</b>	<i>Greedy Randomized Adaptative Search Procedure with Evolutionary Local Search</i>
<b>ILS</b>	<i>Iterated Local Search</i>
<b>IPCs</b>	<i>infeasible-path constraints</i>
<b>L-CVRP</b>	<i>Capacitated Vehicle Routing Problem with Loading Constraints</i>
<b>LMP</b>	<i>Linear Master Problem</i>
<b>MP</b>	<i>Master Problem</i>
<b>MP-VRP</b>	<i>Multi-Pile Vehicle Routing Problem</i>
<b>RCPSP</b>	<i>Resource Constrained Project Scheduling Problem</i>
<b>RCPSP-CVRP</b>	<i>Capacitated Vehicle Routing Problem with Resource Constrained Project Scheduling Problem Constraints</i>
<b>RLMP</b>	<i>Restricted Linear Master Problem</i>
<b>SP</b>	<i>set-partitioning</i>
<b>SPPRC</b>	<i>Shortest Path Problem with Resource Constraints</i>
<b>TSP</b>	<i>Traveling Salesman Problem</i>
<b>VRP</b>	<i>Vehicle Routing Problem</i>
<b>VRPB</b>	<i>Vehicle Routing Problem with Backhauls</i>
<b>VRPPD</b>	<i>Vehicle Routing Problem with Pick-ups and Deliveries</i>
<b>VRPTW</b>	<i>Vehicle Routing Problem with Time Windows</i>

# Sumário

Agradecimentos	ix
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
Lista de Algoritmos	xix
Lista de Siglas	xxi
<b>1 Introdução</b>	<b>1</b>
1.1 Definição do <b>2L-CVRP</b> . . . . .	2
<b>2 Trabalhos Relacionados</b>	<b>7</b>
2.1 Problemas de Roteamento de Veículos . . . . .	7
2.2 Problemas de Corte e Empacotamento Bidimensional . . . . .	8
2.3 Problemas de Roteamento de Veículos com Restrições de Carregamento . . . . .	10
<b>3 Modelos Matemáticos para o <b>2L-CVRP</b></b>	<b>15</b>
3.1 Modelo de Fluxo de Veículo . . . . .	15
3.2 Modelo de Particionamento de Conjuntos . . . . .	17
<b>4 Métodos para Resolução do Carregamento</b>	<b>19</b>
4.1 Heurísticas Construtivas . . . . .	19
4.2 <i>Branch-and-Bound</i> . . . . .	24
4.3 <i>Branch-and-Bound</i> Aproximado . . . . .	28
4.4 Acelerando a Verificação com uma Árvore de Viabilidade . . . . .	28
4.4.1 Inserção na Árvore . . . . .	29
4.4.2 Recuperação da Informação na Árvore . . . . .	30

<b>5</b>	<b>Métodos para Resolução do 2L-CVRP</b>	<b>33</b>
5.1	Heurísticas Baseadas em Geração de Colunas . . . . .	33
5.1.1	Geração do <i>Pool</i> Inicial de Colunas . . . . .	35
5.1.2	O Subproblema de Precificação e a Geração de Novas Colunas . . . . .	42
5.1.3	Heurística de Geração de Colunas com Verificação da Viabilidade dos Carregamentos na Precificação (CGV) . . . . .	46
5.1.4	Heurística de Geração de Colunas sem Verificação da Viabilidade dos Carregamentos na Precificação (CGN) . . . . .	48
5.2	<i>Branch-and-Cut</i> . . . . .	49
<b>6</b>	<b>Experimentos Computacionais</b>	<b>53</b>
6.1	Ambiente Computacional, Instâncias de Teste e Características Consideradas	53
6.2	Configuração dos Algoritmos . . . . .	56
6.3	Ganho de Desempenho com a Árvore de Viabilidade . . . . .	56
6.4	Desempenho dos Algoritmos Desenvolvidos . . . . .	57
6.4.1	Resultados para as <i>Características Ti</i> . . . . .	57
6.4.2	Resultados para as <i>Características Tr</i> . . . . .	61
6.5	Comentários sobre os Resultados . . . . .	66
<b>7</b>	<b>Conclusão e Trabalhos Futuros</b>	<b>69</b>
	<b>Referências Bibliográficas</b>	<b>73</b>
	<b>Apêndice A Resultados Detalhados</b>	<b>79</b>
A.1	Resultados para as Instâncias com <i>Características Ti</i> . . . . .	79
A.2	Resultados para as Instâncias com <i>Características Tr</i> . . . . .	93



# Capítulo 1

## Introdução

O Problema de Roteamento de Veículos (**VRP**, do inglês *Vehicle Routing Problem*) é um dos problemas mais estudados em otimização combinatória. Tal problema é  $\mathcal{NP}$ -difícil e consiste na determinação do melhor conjunto de rotas a serem percorridas por uma frota de veículos, a partir de um depósito central, de modo a satisfazer a demanda de um determinado conjunto de clientes. A versão mais conhecida desse problema é o **VRP** Capacitado (**CVRP**, do inglês *Capacitated Vehicle Routing Problem*), o qual define que os veículos são idênticos e possuem uma capacidade de carga máxima, geralmente representada pelo peso ou volume.

Na maioria dos casos, assume-se que a capacidade dos veículos é representada por uma simples medida unidimensional. No entanto, em muitas aplicações reais, essa abordagem não é adequada, pois a forma como os itens demandados pelos clientes são distribuídos no compartimento do veículo pode ser difícil ou, até mesmo, impossível. Com isso, o **CVRP** com Restrições de Carregamento (**L-CVRP**, do inglês *Capacitated Vehicle Routing Problem with Loading Constraints*) foi proposto, para que o problema de posicionar os itens nos veículos fosse tratado durante a resolução do problema de roteamento. De modo geral, esse problema proíbe que os itens dos clientes se sobreponham dentro do compartimento do veículo.

As restrições consideradas pelo **L-CVRP** podem variar de acordo com as necessidades do problema, contudo, algumas delas são mais comuns. Normalmente, o carregamento dos itens é realizado somente por um dos lados do compartimento do veículo. Por isso, uma restrição de sequencialidade é comumente associada ao posicionamento dos itens, de tal modo a proibir que itens que serão entregues mais tarde por um veículo, obstruam o caminho de descarregamento daqueles que serão entregues antes. Essa restrição é utilizada para que não sejam necessários rearranjos dos itens no compartimento do veículo durante a entrega no cliente, o que proporciona mais rapidez e agilidade nos serviços. Também é frequente que, devido à estrutura do palete ou à distribuição de peso não uniforme, os itens não podem ser manuseados a partir de qualquer lado por equipamentos

de carga e descarga, como por exemplo, empilhadeiras. Por essas razões, restrições que definem se os itens podem ou não ser rotacionados são consideradas.

Neste trabalho, foi considerada a variante bidimensional do **L-CVRP**, denominada **2L-CVRP**, do inglês *Capacitated Vehicle Routing Problem with Two-dimensional Loading Constraints*. Isso significa que os veículos dispõem de compartimentos retangulares para o carregamento, que devem ser preenchidos por itens retangulares de variadas dimensões demandados pelos clientes. O **2L-CVRP** ocorre na prática quando os itens não podem ser empilhados uns sobre os outros, devido às suas fragilidades ou formas. Esse problema é  $\mathcal{NP}$ -difícil, pois trata-se de uma combinação do **CVRP** e do Problema de *Bin Packing* Bidimensional (**2BPP**, do inglês *Two-dimensional Bin Packing Problem*) em um único compartimento, que também são  $\mathcal{NP}$ -difíceis. Os detalhes sobre o **2L-CVRP** são explicados na Seção 1.1.

## 1.1 Definição do 2L-CVRP

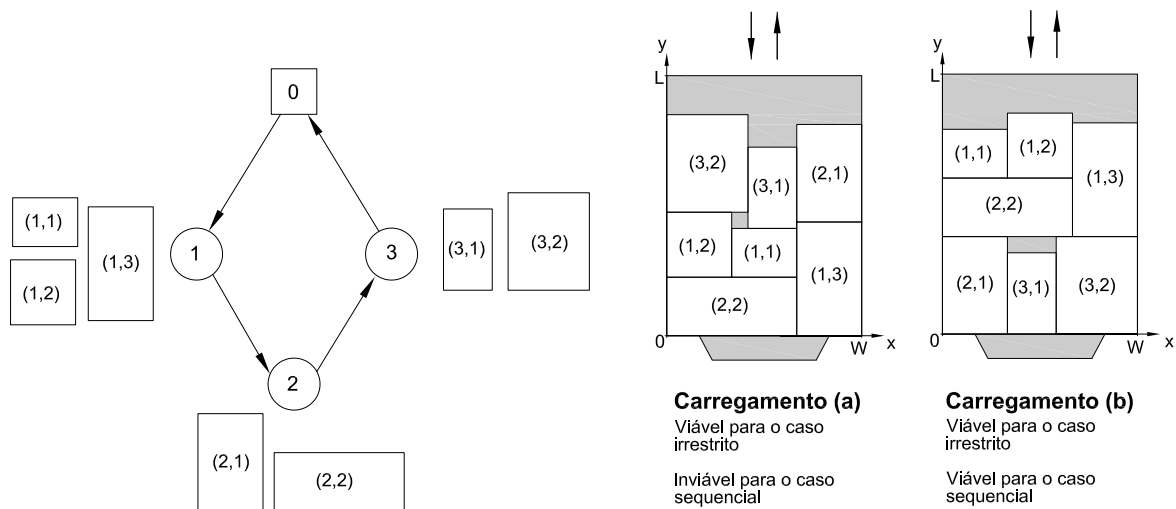
A definição do **2L-CVRP**, descrita por Iori et al. [2007], pode ser expressa da seguinte forma: Seja  $G = (V, E)$  um grafo completo não direcionado, onde  $V$  define o conjunto de  $n + 1$  vértices correspondentes ao depósito (vértice 0) e aos clientes (vértices  $1, \dots, n$ ), e  $E = \{(i, j) \mid i, j \in V\}$  define o conjunto de arestas. Para cada aresta  $(i, j) \in E$  é associado um custo  $c_{ij} \geq 0$ . Um conjunto de  $K$  veículos idênticos são disponibilizados no depósito, cada veículo com uma capacidade em peso  $D$  e um compartimento retangular de área total  $A = WL$ , onde  $W$  é a largura e  $L$  é o comprimento deste compartimento. Além disso, as operações de carregamento e descarregamento são feitas por um único lado do compartimento do veículo.

Em cada cliente  $i$  ( $i \in V \setminus \{0\}$ ) demanda-se um conjunto de  $m_i$  itens retangulares, com peso total  $d_i$ , e cada item tem a largura e o comprimento iguais a  $w_{it}$  e  $l_{it}$  ( $t = 1, \dots, m_i$ ), respectivamente. Dessa forma, cada item é denotado por um par  $(i, t)$ , onde  $i$  representa o cliente e  $t$  o índice do item deste cliente. Além disso, a área total dos itens do cliente  $i$  é denotada por  $a_i = \sum_{t=1}^{m_i} w_{it}l_{it}$ . Assume-se que todos estes valores são inteiros positivos.

As restrições gerais do **2L-CVRP** podem ser definidas como segue: (a) o número de rotas não deve exceder o número de veículos disponíveis, (b) as capacidades em peso e área de cada veículo não podem ser extrapoladas, (c) toda rota começa e termina no depósito, (d) cada cliente é visitado apenas uma vez e por apenas um veículo, (e) os itens tem uma orientação fixa (não podem ser rotacionados) e devem ser carregados com os seus lados paralelos aos lados do compartimento do veículo (ou seja, é necessário um carregamento ortogonal), e (f) os itens devem ser carregados de modo a não haver sobreposição entre eles.

Nesta dissertação são consideradas duas variantes do problema: o 2L-CVRP irrestrito e o 2L-CVRP sequencial. O primeiro não requer restrições adicionais além das especificadas anteriormente, enquanto o segundo necessita de uma restrição de carregamento sequencial, a qual define o posicionamento dos itens no compartimento do veículo de acordo com a ordem em que os clientes são visitados. Essa restrição define que as operações de descarregamento dos itens, que são executadas por um lado do compartimento do veículo, não devem resultar no rearranjo dos itens no veículo ou em deslocamentos laterais dos itens a serem entregues, ou seja, somente movimentos perpendiculares à saída do compartimento são permitidos. Em outras palavras, dado um cliente visitado por um veículo qualquer, seus itens não devem ter a saída obstruída por itens de clientes que serão visitados mais tarde pelo mesmo veículo.

A Figura 1.1 ilustra dois carregamentos possíveis dos itens referentes aos clientes de uma mesma rota do 2L-CVRP. A rota especificada é composta pelos clientes 1, 2 e 3, onde o primeiro demanda os itens (1, 1), (1, 2) e (1, 3), o segundo demanda os itens (2, 1) e (2, 2), e o terceiro demanda os itens (3, 1) e (3, 2). O Carregamento (a) representa uma disposição viável para o caso irrestrito, mas viola a restrição adicional para o caso sequencial, pois os itens (3, 2) e (3, 1) obstruem o descarregamento dos itens (1, 2), (1, 1) e (2, 2), e o item (2, 1) obstrui a saída do item (1, 3). Por outro lado, o Carregamento (b) é viável para os casos irrestrito e sequencial. Nesse caso, os itens (1, 1), (1, 2) e (1, 3) são descarregados inicialmente, seguidos por (2, 2) e (2, 1), e finalmente por (3, 1) e (3, 2).



**Figura 1.1:** Carregamento irrestrito e sequencial.

Para cada rota viável do 2L-CVRP é associado um subconjunto  $S$  de clientes, acompanhado pela ordem em que estes são visitados, dada por uma bijeção  $\sigma : S \rightarrow \{1, \dots, |S|\}$ , onde  $\sigma(i)$  é a ordem que o cliente  $i \in S$  é visitado na rota. Com isso, pode-se definir formalmente as quatro características de uma rota viável para o problema. A primeira delas refere-se à capacidade relativa aos pesos dos itens da rota  $(S, \sigma)$ , que é uma restrição clássica do CVRP:

**Característica 1.** A capacidade do veículo não é violada, ou seja,

$$\sum_{i \in S} d_i \leq D.$$

As próximas características são referentes ao carregamento. Para defini-las, é conveniente fazer o mapeamento da superfície do compartimento do veículo no primeiro quadrante do sistema de coordenadas cartesiano. Dessa forma, o ponto  $(0, 0)$  corresponde ao canto inferior esquerdo da superfície do compartimento do veículo, e o eixo  $x$  e  $y$  representam o fundo (lado referente a onde está a cabine do veículo) e o lado esquerdo do compartimento, respectivamente. A posição de um item  $(i, t)$  no compartimento pode ser definida por duas variáveis  $x_{it}$  e  $y_{it}$ , representando onde o ponto inferior esquerdo do item foi posicionado. O carregamento e descarregamento é feito pelo lado traseiro do compartimento do veículo, localizado em  $y = L$ . Com isso, as seguintes características estão presentes numa rota viável:

**Característica 2.** Os itens estão completamente contidos no compartimento do veículo, ou seja,

$$0 \leq x_{it} \leq W - w_{it} \quad \text{e} \quad 0 \leq y_{it} \leq L - l_{it}$$

para todo  $i \in S$  e  $t \in \{1, \dots, m_i\}$ .

**Característica 3.** Dois itens não se sobrepõem, ou seja,

$$x_{it} + w_{it} \leq x_{jt'} \quad \text{ou} \quad x_{jt'} + w_{jt'} \leq x_{it} \quad \text{ou}$$

$$y_{it} + l_{it} \leq y_{jt'} \quad \text{ou} \quad y_{jt'} + l_{jt'} \leq y_{it}$$

para todo  $\{i, j\} \in S$ ,  $t \in \{1, \dots, m_i\}$ ,  $t' \in \{1, \dots, m_j\}$ , e  $(i, t) \neq (j, t')$ .

Por fim, apresenta-se a Característica 4, que somente é necessária em rotas viáveis do **2L-CVRP** sequencial.

**Característica 4.** Um item  $(i, t)$  não tem o caminho obstruído, mesmo que parcialmente, por um item de um cliente que será atendido mais tarde na rota, ou seja,

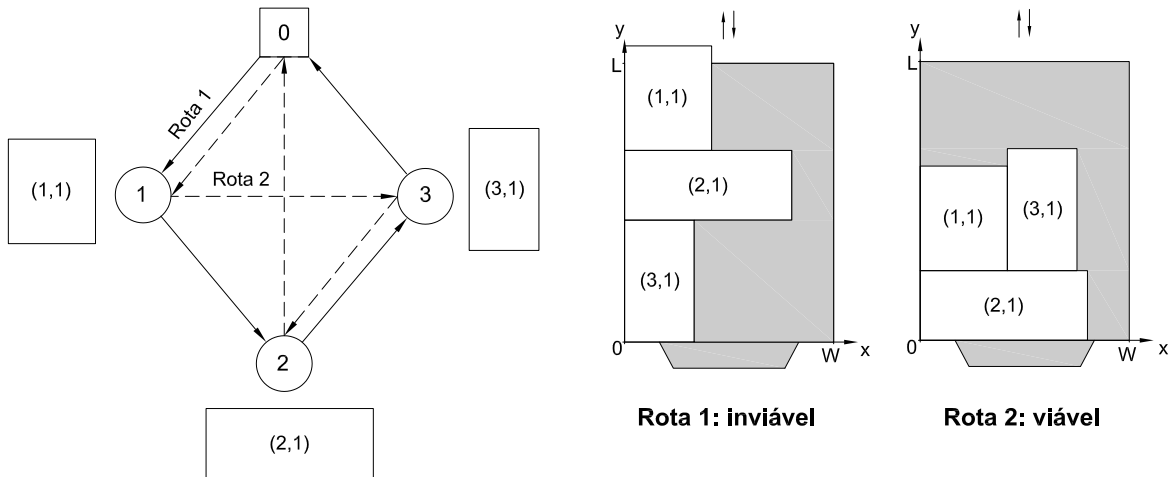
$$y_{jt'} + l_{jt'} \leq y_{it} \quad \text{ou} \quad x_{it} + w_{it} \leq x_{jt'} \quad \text{ou} \quad x_{jt'} + w_{jt'} \leq x_{it}$$

para todo  $\{i, j\} \in S : \sigma(i) < \sigma(j)$ ,  $t \in \{1, \dots, m_i\}$ , e  $t' \in \{1, \dots, m_j\}$ .

O custo associado a uma rota viável  $(S, \sigma)$  é igual a soma dos custos das arestas utilizadas para visitar os clientes em  $S$  de acordo com a ordem  $\sigma$ . Com isso, o objetivo do **2L-CVRP** consiste em encontrar  $K$  rotas viáveis em que o custo total seja mínimo.

Uma observação importante quanto ao **2L-CVRP** sequencial, notada por [Iori et al. \[2007\]](#), é o fato de que a sequência  $\sigma$  é relevante para a viabilidade de uma rota  $(S, \sigma)$ . A Figura 1.2 ilustra duas sequências diferentes  $\sigma^1$  e  $\sigma^2$  associadas ao mesmo subconjunto  $S = \{1, 2, 3\}$ . Cada cliente possui somente um item. A rota  $(S, \sigma^1)$  é definida por  $\sigma^1(1) = 1$ ,  $\sigma^1(2) = 2$  e  $\sigma^1(3) = 3$ , e é inviável, pois o item  $(2, 1)$ , que não pode ficar na

frente do item (1, 1) e nem atrás do item (3, 1), é muito largo para ser posicionado ao lado dos mesmos. Em contrapartida, a rota  $(S, \sigma^2)$ , definida por  $\sigma^2(1) = 1$ ,  $\sigma^2(2) = 3$  e  $\sigma^2(3) = 2$ , é uma rota viável, pois não há obstruções e todos os itens podem ser carregados sem sobreposição.



**Figura 1.2:** Rota inviável e rota viável para o 2L-CVRP sequencial, considerando os mesmos clientes.

Ainda em Iori et al. [2007], os autores observaram que uma rota  $(S, \sigma^{2R})$ , no sentido oposto à  $\sigma^2$ , definida por  $\sigma^{2R}(1) = 3$ ,  $\sigma^{2R}(2) = 1$  e  $\sigma^{2R}(3) = 2$ , também é uma rota viável para o 2L-CVRP. De um modo geral, se  $(S, \sigma)$  é uma rota viável, então  $(S, \sigma^R)$ , onde  $\sigma^R = |S| - \sigma(i) + 1$ , também é viável. O carregamento da rota contrária pode ser obtido ao aplicar uma operação de simetria em relação a linha  $y = L/2$ , e ao mudar, consequentemente, as coordenadas  $(x, y)$  para  $(x^R, y^R)$ , tal que  $x_{it}^R = x_{it}$  e  $y_{it}^R = L - y_{it} - l_{it}$ . De forma análoga, a rota contrária de uma rota inviável também é inviável.



# Capítulo 2

## Trabalhos Relacionados

Este capítulo apresenta uma revisão bibliográfica sobre as versões do **VRP**, as variações relacionadas aos Problemas de Empacotamento Bidimensional (**2CPP**, do inglês *Two-dimensional Cutting and Packing Problems*), e os Problemas de Roteamento de Veículos com Restrições de Carregamento (**L-CVRP**). Na Seção 2.1 são apresentadas diferentes versões do **VRP**. Por sua vez, os **2CPP** são exibidos na Seção 2.2. Finalmente, na Seção 2.3, os Problemas de Roteamento de Veículos com Restrições de Carregamento são apresentados, onde se enquadra o **2L-CVRP**.

### 2.1 Problemas de Roteamento de Veículos

O Problema de Roteamento de Veículos, em sua primeira versão, foi proposto por **Dantzig & Ramser [1959]**. Os autores definiram o primeiro modelo para um problema de planejamento de transportes. Por ser uma generalização do Problema do Caixeiro Viajante (**TSP**, do inglês *Traveling Salesman Problem*), o **VRP** também é classificado como um problema  $\mathcal{NP}$ -difícil. A partir desse trabalho, várias modificações do problema foram propostas, de modo a tentar deixá-lo mais próximo da realidade.

A versão mais básica do **VRP** é o Problema de Roteamento de Veículos Capacitados (**CVRP**). Este problema consiste em clientes com demandas conhecidas, que são atendidos por veículos idênticos que pertencem ao mesmo depósito. Formalmente, o problema pode ser definido como um grafo completo  $G = (V, E)$ , onde  $V = \{0, \dots, n\}$  é o conjunto de nós, tal que 0 representa o depósito e  $1, \dots, n$  representam os clientes, e  $E = \{(i, j) \mid i, j \in V\}$  é o conjunto de arestas. Para cada arco  $(i, j) \in E$  é associado um custo não negativo  $c_{ij}$ , representando o custo de deslocamento de  $i$  para  $j$ . Além disso, cada cliente  $i \in V$  tem uma demanda  $d_i$  não negativa, e os veículos têm uma capacidade limitada  $Q$ . O **CVRP** tem por objetivo definir rotas para cada um dos veículos de forma que a soma de seus custos seja mínima, respeitando as restrições: (a) a capacidade do veículo não deve ser excedida em nenhuma das rotas, (b) cada cliente deve ser visitado somente uma

vez e por um único veículo, e (c) cada rota deve começar e terminar no depósito.

Muitas abordagens foram desenvolvidas, ao decorrer dos anos, para resolver o **CVRP**. A primeira delas foi uma heurística gulosa proposta por [Clarke & Wright \[1964\]](#), conhecida como algoritmo de *savings* de Clarke-Wright. A partir de então, muitos métodos heurísticos e exatos surgiram. O livro editado por [Toth & Vigo \[2001b\]](#) reúne abordagens heurísticas clássicas ([Laporte & Semet \[2001\]](#)), metaheurísticas ([Gendreau et al. \[2001\]](#)) e abordagens exatas baseadas em *Branch-and-Bound* (**BB**) ([Toth & Vigo \[2001a\]](#)), *Branch-and-Cut* (**BC**) ([Naddef & Rinaldi \[2001\]](#)) e *Branch-and-Price* (**BP**) ([Bramel & Simchi-Levi \[2001\]](#)). Em uma abordagem hibridizada mais recente, [Fukasawa et al. \[2006\]](#) propõem um *Branch-and-Cut-and-Price* (**BCP**) Robusto que consegue resolver instâncias com até 100 clientes.

Além do **CVRP**, outras versões do **VRP** são muito estudadas. O Problema de Roteamento de Veículos com Janelas de Tempo (**VRPTW**, do inglês *Vehicle Routing Problem with Time Windows*) é uma extensão do **CVRP** onde cada cliente deve ter seu atendimento iniciado em uma janela de tempo e o veículo associado deve atendê-lo durante um tempo previamente estipulado. Por sua vez, o Problema de Roteamento de Veículos com *Backhauls* (**VRPB**, do inglês *Vehicle Routing Problem with Backhauls*) consiste em um **CVRP** onde o conjunto de clientes é particionado em dois subconjuntos: *linehaul* e *backhaul*. O primeiro subconjunto consiste nos clientes que necessitam de itens a serem entregues, enquanto o segundo representa os clientes que dispõem de itens a serem coletados. No **VRPB**, todos os clientes *linehaul* devem ser visitados antes dos clientes *backhaul*. Uma outra variação do **VRP** é o Problema de Roteamento de Veículos com Coleta e Entrega (**VRPPD**, do inglês *Vehicle Routing Problem with Pick-ups and Deliveries*), onde uma requisição de transporte é associada a dois clientes, de tal forma que a demanda é coletada em um deles e entregue no outro. Nesse problema, uma solução viável requer que a coleta de uma requisição seja feita antes de sua entrega, e que ambas operações ocorram na mesma rota. Informações sobre os trabalhos propostos e os detalhes do **VRPTW**, **VRPB** e **VRPPD**, podem ser encontrados em [Alvarenga et al. \[2007\]](#), [Toth & Vigo \[2001c\]](#) e [Desaulniers et al. \[2001\]](#), respectivamente.

## 2.2 Problemas de Corte e Empacotamento Bidimensional

No final da década de 1930, Kantorovich publicou o primeiro trabalho relacionado aos Problemas de Corte e Empacotamento Bidimensional (**2CPP**). Seu trabalho, inicialmente publicado em russo, foi traduzido posteriormente para o inglês ([Kantorovich \[1960\]](#)). Apesar de sua contribuição expressiva, do ponto de vista matemático, somente pequenos problemas podiam ser resolvidos. A dificuldade de resolução dos **2CPP** se deve ao fato de



serem  $\mathcal{NP}$ -difíceis, pois são generalizações do Problema de Corte de Padrões Unidimensional (1CSP, do inglês *One-dimensional Cutting Stock Problem*) (Gilmore & Gomory [1961], Gilmore & Gomory [1963]), que também é  $\mathcal{NP}$ -difícil.

De modo geral, os 2CPP consistem em problemas onde itens retangulares menores devem preencher um compartimento retangular maior (característica dos problemas de empacotamento) ou, de maneira análoga, onde uma peça retangular maior deve ser cortada em peças retangulares menores (característica dos problemas de corte). Formalmente, os problemas de empacotamento (corte) podem ser definidos como um conjunto de itens  $I = \{1, \dots, n\}$  que devem ser empacotados (cortados) em um ou mais compartimentos (a partir de uma ou mais peças)  $B = (W, L)$ , onde  $W$  é a largura e  $L$  o comprimento. Um item  $i$  tem uma largura  $w_i \in \mathbb{N}$  e um comprimento  $l_i \in \mathbb{N}$ . Nesses problemas, os itens devem ser empacotados (cortados) de tal forma que (a) não haja sobreposição entre eles, e (b) não haja extrapolação dos limites de  $B$ . Os objetivos mais comuns são: minimizar o número de compartimentos (peças) utilizados (ex. Problema de *Bin Packing* Bidimensional (2BPP), Problema de Corte de Padrões Bidimensional (2CSP, do inglês *Two-dimensional Cutting Stock Problem*)) e maximizar o número de itens empacotados (cortados) (ex. Problema da Mochila Bidimensional (2D-KP, do inglês *Two-dimensional Knapsack Problem*)).

Gilmore & Gomory [1965] propuseram a primeira abordagem de sucesso para resolver um dos 2CPP. Eles propuseram um método de decomposição por geração de colunas para o 2CSP. A partir desse trabalho, a formulação inicial sofreu várias alterações de modo a obter melhorias. Além disso, muitas variantes do problema surgiram, fazendo necessária uma tipologia bem elaborada para classificar os problemas existentes e aqueles que ainda iriam surgir. Dyckhoff [1990] propôs a primeira tipologia para os Problemas de Corte e Empacotamento, que serviu como um excelente instrumento de organização e categorização da literatura. Contudo, no decorrer dos anos foram observadas algumas deficiências nessa tipologia, que contribuíram para a criação de um novo padrão por Wäscher et al. [2007], que é o trabalho mais recente nesse aspecto.

Dentre os 2CPP existentes na literatura, os que mais se aproximam do problema de empacotamento do 2L-CVRP em estudo são: o Problema de *Bin Packing* Bidimensional (2BPP), o Problema de *Strip Packing* Bidimensional (2SPP, do inglês *Two-dimensional Strip Packing Problem*), e o Problema de Empacotamento Ortogonal Bidimensional (2OPP, do inglês *Two-dimensional Orthogonal Packing Problem*). O 2BPP consiste em empacotar um conjunto de itens retangulares em um número mínimo de compartimentos retangulares idênticos. Por outro lado, o 2SPP consiste em empacotar um conjunto de itens retangulares em uma fita (*strip*) de largura conhecida e comprimento infinito, tal que o objetivo é minimizar o comprimento total do empacotamento. Finalmente, o 2OPP consiste em determinar se um conjunto de itens retangulares pode ser, ou não, empacotado em um compartimento retangular de tamanho fixo. Esse último é o que mais se

assemelha ao problema de empacotamento do **2L-CVRP**, pois o objetivo é apenas verificar se existe uma disposição viável dos itens. Uma revisão bibliográfica mais aprofundada sobre estes e outros **2CPP** pode ser encontrada em [Dowland & Dowland \[1992\]](#) e [Lodi et al. \[2002\]](#).

## 2.3 Problemas de Roteamento de Veículos com Restrições de Carregamento

A combinação entre o **VRP** e os Problemas de Carregamento surgiu em trabalhos recentes. Pelo que se conhece, o primeiro problema resultante desta combinação foi proposto por [Iori \[2004\]](#), onde o autor combinou o **CVRP** e uma variação dos **2CPP**, dando origem ao **2L-CVRP**. Os trabalhos relacionados a esse e a outras variantes do problema estão relacionados a seguir.

[Iori et al. \[2007\]](#) apresentaram um algoritmo **BC** baseado na formulação clássica de fluxo de veículo para resolver o **2L-CVRP** sequencial. Esse **BC** usa desigualdades válidas da literatura do **CVRP** e também desigualdades específicas associadas às sequências com carregamento inválido. A viabilidade dos carregamentos das rotas é desconsiderada até que seja encontrada uma solução inteira pelo **BC**. Quando isso ocorre, o problema de carregamento associado é resolvido invocando um **BB**. Se algum carregamento for inviável, um corte de inviabilidade de rota é adicionado. Caso contrário, a solução deve ser atualizada. Esta abordagem permitiu que algumas instâncias com até 35 clientes e mais de 100 itens fossem resolvidas.

Devido ao fato da abordagem proposta por [Iori et al. \[2007\]](#) ter resolvido somente instâncias pequenas, heurísticas foram propostas a partir de então para que instâncias maiores e, portanto, mais próximas de casos do mundo real, pudessem ser resolvidas. A primeira heurística para resolver o **2L-CVRP** sequencial e irrestrito foi proposta por [Gendreau et al. \[2008\]](#). Os autores propuseram um método Busca Tabu, onde sua vizinhança de busca consiste nas soluções que diferem de uma outra solução pela troca de um cliente de uma rota para outra. Uma vez que um movimento é realizado nessa vizinhança, ele é inserido na lista tabu. Além disso, movimentos que resultam em soluções inviáveis são permitidos, porém, são adicionadas penalidades às suas funções objetivo, proporcionalmente ao nível de violação das restrições do problema. Os resultados obtidos por essa heurística se mostraram competitivos, conseguindo alcançar soluções ótimas em 50% das instâncias que [Iori et al. \[2007\]](#) atestaram a otimalidade.

[Zachariadis et al. \[2009\]](#) apresentaram uma Busca Tabu Guiada para resolver o **2L-CVRP** sequencial e irrestrito. Esse procedimento tem uma estratégia de busca similar a uma Busca Tabu, todavia a função objetivo é alterada de modo a aumentar a diversificação. Para tentar definir se um carregamento é viável, usou-se um conjunto com cinco

heurísticas construtivas. Para acelerar a convergência do algoritmo, reduziu-se o tamanho da vizinhança explorada a cada iteração, e foi inserida uma estrutura que armazena informações sobre a viabilidade dos carregamentos das rotas já testadas, como forma de não precisar testá-los novamente. Esta abordagem obteve resultados que superaram a maioria dos resultados da literatura.

Fuellerer et al. [2009] resolveram pela primeira vez o 2L-CVRP não-orientado, ou seja, os itens podem ser rotacionados em 90° para serem carregados. Os autores utilizaram um algoritmo de Otimização de Colônia de Formigas, que é uma extensão do mesmo algoritmo para o CVRP. O método utilizado busca sempre gerar rotas viáveis para os veículos. Para garantir esta viabilidade, são usados limites inferiores, heurísticas e um *Branch-and-Bound* limitado no número de iterações e no tempo de execução, de forma que as heurísticas são utilizadas somente se os limites inferiores não provarem inviabilidade, e o BB truncado só é usado se as heurísticas falharem. As combinações de clientes em uma rota feita pelo algoritmo levam em conta regras probabilísticas baseadas no algoritmo de *savings* de Clarke-Wright e informações dos feromônios. Os resultados obtidos superaram os dos trabalhos de Gendreau et al. [2008] e Zachariadis et al. [2009] na média.

Ainda no mesmo ano (2009), Azevedo et al. [2009] e Azevedo [2009] propuseram um algoritmo BC para resolver o 2L-CVRP sequencial e irrestrito. Nessa abordagem, foram adaptadas e utilizadas sete desigualdades válidas para o CVRP. Por sua vez, a viabilidade do carregamento é garantida através de uma nova heurística desenvolvida e de um algoritmo exato. Comparado ao trabalho de Iori et al. [2007], esta estratégia obteve nove novas soluções ótimas para o caso sequencial e o tempo de resolução foi melhor para várias instâncias. Além disso, no caso irrestrito, que ainda não havia sido tratado por um método exato, cinquenta instâncias foram resolvidas na otimalidade.

Leung et al. [2010] apresentaram um algoritmo *Simulated Annealing* para resolver o 2L-CVRP sequencial e irrestrito. Para definir a viabilidade dos carregamentos, os autores utilizaram o conjunto de heurísticas definido por Zachariadis et al. [2009] e mais uma nova heurística. Essa nova heurística posiciona os itens com base em um *fitness*, que define qual a melhor posição a serem colocados. Além disso, foi usada uma árvore *Trie* para armazenar quais rotas são viáveis ou não, o que acelera a execução do algoritmo. Esse algoritmo obteve poucos resultados melhores que os resultados de Fuellerer et al. [2009].

Leung et al. [2011] propuseram uma Busca Tabu Guiada Estendida para resolver o 2L-CVRP sequencial e irrestrito. Esse método baseia-se na Busca Tabu Guiada de Zachariadis et al. [2009]. Contudo, na versão estendida, um movimento é aceito se ele produz uma nova solução incumbente, mesmo que ele esteja na lista tabu. Para atestar a viabilidade dos carregamentos das rotas, essa abordagem utiliza do mesmo conjunto de heurísticas utilizado em Leung et al. [2010]. Os resultados obtidos por Leung et al. [2011] superaram aqueles obtidos por Zachariadis et al. [2009].

Por último, Duhamel et al. [2011] apresentaram uma Busca Local Evolutiva com

Múltiplas Inicializações para resolver o **2L-CVRP** irrestrito. Esse método consiste em duas etapas. Na primeira etapa, as restrições de carregamento são relaxadas de modo a transformá-las em restrições do Problema de Sequenciamento de Projetos com Restrição de Recurso (**RCPSP**, do inglês *Resource Constrained Project Scheduling Problem*), resultando num **CVRP** com restrições do **RCPSP**, ou **RCPSP-CVRP**, do inglês *Capacitated Vehicle Routing Problem with Resource Constrained Project Scheduling Problem Constraints*, que trata-se de um problema com resolução mais fácil. A segunda etapa consiste em converter a solução do **RCPSP-CVRP** em uma solução do **2L-CVRP**. O algoritmo usado para resolver o problema da primeira etapa combina um *Greedy Randomized Adaptive Search Procedure* (**GRASP**) com um método de Busca Local Evolutiva, denominado **GRASP x ELS**, do inglês *Greedy Randomized Adaptive Search Procedure with Evolutionary Local Search*. O método se mostrou muito eficiente, uma vez que superou todos os melhores resultados publicados anteriormente.

Na Tabela 2.1 são exibidas algumas características que foram adotadas pelos trabalhos supracitados ao resolver o **2L-CVRP**. Essas características são importantes para que se possa saber com qual trabalho se deve comparar os resultados. A primeira coluna da tabela contém a referência do trabalho em questão. A segunda coluna indica se o carregamento é sequencial e orientado (SO), irrestrito e orientado (UO), sequencial e não-orientado (SN) ou irrestrito e não-orientado (UN). A terceira coluna exibe se o número de rotas tem que ser igual ao número total de veículos ( $= |K|$ ), ou se pode ser menor ( $\leq |K|$ ). A quarta coluna mostra se são permitidas rotas com somente um cliente ( $\checkmark$ ), ou não ( $-$ ). A última coluna apresenta se os custos considerados entre os clientes são os valores reais ou inteiros, sendo este último por decorrência de alguma operação matemática (ex. truncamento, arredondamento).

**Tabela 2.1:** Características adotadas pelos trabalhos da literatura na resolução do **2L-CVRP**

Trabalho	Tipo de carregamento	Número de rotas	Rota com um cliente	Custo
Iori et al. [2007]	SO	$=  K $	$-$	truncado
Gendreau et al. [2008]	SO	$=  K $	$-$	truncado
	SO/UO	$\leq  K $	$\checkmark$	real
Zachariadis et al. [2009]	SO/UO	$\leq  K $	$\checkmark$	real
Fuellerer et al. [2009]	SO	$=  K $	$-$	truncado
	SO/UO/SN/UN	$\leq  K $	$\checkmark$	real
Azevedo et al. [2009]	SO/UO	$=  K $	$-$	truncado
Leung et al. [2010]	SO/UO	$\leq  K $	$\checkmark$	real
Leung et al. [2011]	SO/UO	$\leq  K $	$\checkmark$	real
Duhamel et al. [2011]	UO	$\leq  K $	$\checkmark$	real

Uma extensão do **2L-CVRP** é o Problema de Roteamento de Veículos Capacitados com Restrições de Carregamento Tridimensional (**3L-CVRP**, do inglês *Capacitated Vehicle Routing Problem with Three-dimensional Loading Constraints*), originado no trabalho

de [Gendreau et al. \[2006\]](#). Esse problema, ao contrário do **2L-CVRP**, representa o caso em que os itens carregados nos veículos podem ser empilhados. Todos os métodos encontrados na literatura para a resolução do **3L-CVRP** são heurísticos. Isso se deve ao fato que a resolução do problema de carregamento tridimensional é um problema muito difícil de ser resolvido de forma exata em um tempo razoável. As abordagens propostas são: Busca Tabu ([Gendreau et al. \[2006\]](#)), Busca Tabu com método de Cubóides ([Araújo \[2006\]](#)), Busca Tabu Guiada ([Tarantilis et al. \[2009\]](#)), Otimização de Colônia de Formigas ([Fuellerer et al. \[2010\]](#)), Busca Tabu com Busca em Árvore para o carregamento ([Bortfeldt \[2012\]](#)), e *Honey Bee Mating Optimization* ([Ruan et al. \[2013\]](#)).

Outros problemas que combinam o **VRP** com os Problemas de Carregamento ainda são pouco estudados. Dentre esses, pode-se citar o Problema de Roteamento de Veículos com Janelas de Tempo e Restrições de Carregamento Tridimensional (**3L-VRPTW**, do inglês *Vehicle Routing Problem with Time Windows and Three-dimensional Loading Constraints*), que além de ter as características do **VRPTW**, tem também as do **3L-CVRP**. Métodos para resolver este problema são encontrados em [Moura \[2008\]](#) e [Moura & Oliveira \[2009\]](#). Mais um problema nesta categoria é o Problema de Roteamento de Veículos com Múltiplas Pilhas (**MP-VRP**, do inglês *Multi-Pile Vehicle Routing Problem*), que é um caso especial do **3L-CVRP**, onde todos os itens a serem carregados têm a mesma largura, dois tipos de comprimento (pequeno e grande) e altura variável. Esse problema foi proposto por [Doerner et al. \[2007\]](#) para retratar um problema enfrentado por companhias que fazem entregas de produtos de madeira. Outros problemas e trabalhos envolvendo o **VRP** e os Problemas de Carregamento podem ser encontrados em [Wang et al. \[2009\]](#), [Iori & Martello \[2010\]](#) e [Pinto et al. \[2011\]](#).

Neste trabalho, são apresentadas novas abordagens para a resolução do **2L-CVRP** sequencial e irrestrito, ambos com carregamento orientado. No caso, duas heurísticas baseadas em Geração de Colunas (**CG**, do inglês *Column Generation*) foram propostas, e um **BC**, baseado naquele de [Iori et al. \[2007\]](#), foi desenvolvido para tratar as características do problema ainda não tratadas por um método exato (as mesmas adotadas por [Zachariadis et al. \[2009\]](#), mostradas na Tabela 2.1). Pelo que se sabe, não há registro de métodos semelhantes ou que fazem o mesmo na literatura. Com relação à verificação da viabilidade dos carregamentos, foram usados métodos heurísticos e exatos propostos em outros trabalhos ([Zachariadis et al. \[2009\]](#) e [Martello & Vigo \[1998\]](#)). Além disso, uma estrutura de dados para armazenar as viabilidades dos carregamentos já testados foi desenvolvida, baseando-se na árvore *Trie* utilizada por [Leung et al. \[2010\]](#). Essa nova estrutura é capaz de definir a viabilidade de carregamentos ainda não testados, com base naqueles armazenados, proporcionando baixo consumo de memória e eficiência nas verificações.



# Capítulo 3

## Modelos Matemáticos para o 2L-CVRP

Neste capítulo são apresentados dois modelos matemáticos para o 2L-CVRP. Primeiramente, a Seção 3.1 exibe uma reformulação do Modelo de Fluxo de Veículo, proposta por Iori et al. [2007]. Em seguida, a Seção 3.2 apresenta um modelo de Particionamento de Conjuntos, que é usado nos algoritmos baseados em Geração de Colunas desta dissertação. Os detalhes destes modelos são explicados nos itens a seguir.

### 3.1 Modelo de Fluxo de Veículo

Os Modelos de Fluxo de Veículo são os modelos mais usados para as versões básicas do VRP. Suas variáveis de decisão são inteiras e representam o número de vezes que cada arco ou aresta do grafo é utilizado pelos veículos. Particularmente, esses modelos são escolhidos para os casos em que o custo da solução pode ser expresso pela soma dos custos dos arcos ou arestas associados.

Dado um grafo  $G = (V, E)$ ,  $V$  representa o conjunto de vértices e  $E$  o conjunto de arestas. A reformulação de Iori et al. [2007] do Modelo de Fluxo de Veículo utiliza variáveis binárias  $z_{ij}$ , para cada  $(i, j) \in E$ , tal que uma variável assume o valor 1 se e somente se um veículo passa pela aresta  $(i, j)$ . Além disso, dado um subconjunto de clientes  $S$ ,  $r(S)$  representa o número mínimo de veículos necessário para atendê-lo e  $\delta(S)$  denota o conjunto de arestas com uma extremidade em  $S$  e outra em  $V \setminus S$ . Para simplificar a notação,  $\delta(i)$  é usado no lugar de  $\delta(\{i\})$ , tal que  $i \in V$ . Por outro lado, dada uma rota viável  $(S, \sigma)$ ,  $E(S, \sigma)$  denota o conjunto de arestas dessa rota. Já  $\Sigma(S)$  denota a coleção de sequências  $\sigma$  tal que  $(S, \sigma)$  é uma rota viável e  $S$  é um subconjunto de clientes. O modelo resultante é:

$$\min \sum_{(i,j) \in E} c_{ij} z_{ij} \quad (3.1)$$

$$\text{s.a.} \sum_{(i,j) \in \delta(i)} z_{ij} = 2 \quad \forall i \in V \setminus \{0\} \quad (3.2)$$

$$\sum_{(i,j) \in \delta(0)} z_{ij} = 2K \quad (3.3)$$

$$\sum_{(i,j) \in \delta(S)} z_{ij} \geq 2r(S) \quad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (3.4)$$

$$\sum_{(i,j) \in E(S,\sigma)} z_{ij} \leq |S| - 1 \quad \forall (S,\sigma) \text{ tal que } \sigma \notin \Sigma(S) \quad (3.5)$$

$$z_{ij} \in \{0, 1\} \quad \forall (i,j) \in E \quad (3.6)$$

As restrições de grau 3.2 e 3.3 definem o número de arestas incidentes nos vértices clientes e no depósito, respectivamente. As restrições 3.4, conhecidas como *capacity-cut constraints* (CCCs), são responsáveis, juntamente com as restrições de grau, por manter a conectividade e a viabilidade da solução no que diz respeito às capacidades em peso e área dos veículos. De fato, as CCCs estipulam que cada corte  $(S, V \setminus S)$ , definido por um subconjunto  $S$  de clientes, seja cruzado por um número de arestas não inferior a  $r(S)$ . No caso do 2L-CVRP, para  $r(S)$  ser calculado é necessário a resolução de um 2BPP com restrições adicionais que garantem que os itens do mesmo cliente estejam no mesmo compartimento. Por envolver a resolução de um problema  $\mathcal{NP}$ -difícil,  $r(S)$  é substituído por um limite inferior simples  $r'(S)$ . O valor usado é apresentado na Equação 3.7.

$$r'(S) = \max \left\{ \left\lceil \frac{\sum_{i \in S} d_i}{D} \right\rceil, \left\lceil \frac{\sum_{i \in S} a_i}{A} \right\rceil \right\} \quad (3.7)$$

As restrições 3.5, conhecidas como *infeasible-path constraints* (IPCs), são responsáveis por garantir a viabilidade do carregamento dos veículos. Dada uma rota inviável  $(S, \sigma)$ , tais restrições proíbem que essa seja usada, por meio do impedimento de que todas suas arestas sejam selecionadas ao mesmo tempo. Finalmente, as restrições 3.6 impõem que todas as variáveis são binárias, o que implica que todas as rotas devem ter ao menos dois clientes.

Devido ao fato de conter as grandes famílias de restrições 3.4 e 3.5, a resolução do Modelo de Fluxo de Veículo se dá através da utilização de um algoritmo BC. A princípio, com esse método, não é necessário incluir todas essas restrições explicitamente no modelo, porém, ao longo da execução do mesmo, as restrições violadas devem ser explicitadas. Enquanto houverem restrições violadas, procedimentos denominados algoritmos de separação devem identificá-las e explicitá-las. O BC encontra uma solução final para o problema quando esses procedimentos não identificam mais nenhuma restrição violada.



## 3.2 Modelo de Particionamento de Conjuntos

A formulação de Particionamento de Conjuntos (**SP**, do inglês *set-partitioning*) para o **VRP** foi proposta por Balinski & Quandt [1964]. Cada variável de decisão dessa formulação representa uma rota viável para o problema, e o conjunto de todas essas variáveis representa todas as rotas viáveis existentes. Dessa forma, tem-se um número exponencial de variáveis. O fato do modelo ter somente variáveis que representam rotas viáveis faz com que essa formulação detenha, normalmente, limites de relaxação linear mais apertados que a formulação apresentada na Seção anterior.

Nesta dissertação, a mesma formulação proposta por Balinski & Quandt [1964] é utilizada, pois as restrições adicionais do **2L-CVRP** são tratadas apenas no momento da geração das rotas. Portanto, o modelo pode ser definido da mesma forma. Seja  $R$  o conjunto de todas as rotas viáveis para o **2L-CVRP**. Para cada rota  $r \in R$ , tem-se dois parâmetros definidos:  $\alpha_{ir}$  e  $c_r$ . O primeiro assume o valor 1 se a rota  $r$  visita o cliente  $i \in V \setminus \{0\}$  ou o valor 0 caso contrário. Já o último, representa o custo total de  $r$ . Por fim,  $\lambda_r$  é a variável binária de decisão que indica pelo valor 1 se uma rota é utilizada na solução ou por 0 caso contrário. A solução do **SP** consiste em selecionar  $K$  rotas de custo mínimo de tal forma que cada cliente seja visitado exatamente por uma rota. O modelo é apresentado a seguir:

$$\min \sum_{r \in R} c_r \lambda_r \quad (3.8)$$

$$\text{s.a.} \sum_{r \in R} \alpha_{ir} \lambda_r = 1 \quad \forall i \in V \setminus \{0\} \quad (3.9)$$

$$\sum_{r \in R} \lambda_r = K \quad (3.10)$$

$$\lambda_r \in \{0, 1\} \quad \forall r \in R \quad (3.11)$$

As restrições 3.9 impõem que cada cliente  $i$  deve ser visitado exatamente uma vez pelas rotas, enquanto a restrição 3.10 exige que  $K$  rotas sejam selecionadas. Para que seja permitido o uso de menos que  $K$  rotas, basta substituir o sinal de igualdade da restrição 3.10 pelo sinal de menor ou igual. Finalmente, as restrições 3.11 garantem que todas as variáveis são binárias.

Assim como o Modelo de Fluxo de Veículo, o modelo apresentado nesta Seção representa uma formulação não compacta. Nesse caso, porém, o número de variáveis que é exponencial. Apesar de existir uma formulação compacta para o **CVRP**, onde a família exponencial de restrições de eliminação de subciclo é substituída por restrições de *Miller-Tucker-Zemlin* (Miller et al. [1960]), para o **2L-CVRP** ainda não foi encontrada na literatura.

Devido ao número exponencial de variáveis, a resolução do **SP** depende de uma

estratégia baseada em **CG**. O algoritmo de **CG** é capaz de obter um limite inferior para a solução ótima do **SP** através da resolução da relaxação linear do mesmo. Inicialmente, o método necessita apenas de uma solução básica para o problema relaxado e, ao longo de sua execução, novas colunas são geradas implicitamente pela resolução de um sub-problema de precificação. Para a resolução exata do problema inteiro, faz-se necessária a utilização do método **BP**, que equivale a um método **BB** onde cada nó da árvore de enumeração é resolvido por uma **CG**.

# Capítulo 4

## Métodos para Resolução do Carregamento

No **2L-CVRP**, o Problema de Carregamento é um ponto determinante para a obtenção de boas soluções. Se um carregamento associado a uma rota é considerado inviável, essa rota não fará mais parte da solução. Deste modo, se um algoritmo responsável por determinar a viabilidade de um carregamento falhar (não achar uma disposição viável quando esta existir), o custo final da solução pode ser comprometido. Para tentar evitar esse fator indesejável, neste trabalho foram utilizados vários métodos para a verificação da viabilidade. A Seção 4.1 apresenta cinco heurísticas construtivas para fazer a verificação. A Seção 4.2 apresenta um **BB** para resolver o problema na otimalidade, enquanto a Seção 4.3 mostra o mesmo **BB** com algumas restrições que o faz executar em um tempo razoável. Por fim, a Seção 4.4 apresenta uma estrutura de dados desenvolvida para acelerar o algoritmo do **2L-CVRP**, a fim de se evitar que um mesmo carregamento seja analisado duas vezes.

### 4.1 Heurísticas Construtivas

As heurísticas construtivas são métodos utilizados, geralmente, para a obtenção de uma solução inicial com baixo esforço computacional. Esses métodos se baseiam em critérios pré-definidos para a construção de uma solução. Nesta dissertação, cinco heurísticas construtivas foram usadas para a verificação da viabilidade dos carregamentos dos veículos. Tais heurísticas foram utilizadas por [Zachariadis et al. \[2009\]](#) e, basicamente, se fundamentam em diferentes regras para o posicionamento dos itens dentro do compartimento do veículo.

Inicialmente, os itens dos clientes da rota a ser analisada devem ser ordenados. No caso, três ordenações ( $ord_{seq}$ ,  $ord_{Rseq}$ ,  $ord_{unr}$ ) são utilizadas. As ordenações  $ord_{seq}$  e  $ord_{Rseq}$  são mais apropriadas para o **2L-CVRP** sequencial, enquanto a  $ord_{unr}$  é adequada

ao 2L-CVRP irrestrito. A ordenação  $ord_{seq}$  organiza os itens em ordem decrescente com relação a ordem de visitação de seus clientes associados. Em seguida, cada grupo de itens do mesmo cliente é ordenado em ordem decrescente com relação às suas áreas, de tal forma que os empates são quebrados ordenando-os decrescentemente segundo seus comprimentos e larguras, respectivamente. Analogamente, a ordenação  $ord_{Rseq}$  primeiramente inverte a ordem de visitação dos clientes e depois aplica o procedimento da  $ord_{seq}$ . Essa ordenação foi utilizada com o intuito de se aproveitar da propriedade explicitada no final da Seção 1.1, a qual diz que a viabilidade – ou inviabilidade – de uma rota não depende de seu sentido. Já a  $ord_{unr}$  simplesmente organiza os itens de todos os clientes em ordem decrescente segundo suas áreas.

Após ordenados os itens, as heurísticas construtivas os selecionam sucessivamente para carregá-los no compartimento do veículo. Para representar as posições disponíveis para a inserção de novos itens, uma lista denominada  $posList$  é mantida. No início, a única posição disponível é o canto inferior esquerdo do compartimento (o ponto  $(0, 0)$ ). Quando um item é inserido, a posição em que foi colocado é removida da  $posList$  e, no máximo, quatro novas posições são criadas e adicionadas na mesma lista. Dessa forma, é permitido que buracos no carregamento sejam preenchidos posteriormente.

A Figura 4.1 ilustra como é feita a inserção de um item e como as novas posições são geradas. Nesse exemplo, o item  $(1, 5)$  é inserido na posição  $(w_{12}, l_{11})$ . Ao ser inserido, a posição onde foi colocado se torna indisponível e quatro novas posições são criadas. As duas primeiras posições criadas se referem aos pontos inferior direito e superior esquerdo do item inserido, ou seja, as posições  $(w_{12} + w_{15}, l_{11})$  e  $(w_{12}, l_{11} + l_{15})$ , respectivamente. Por

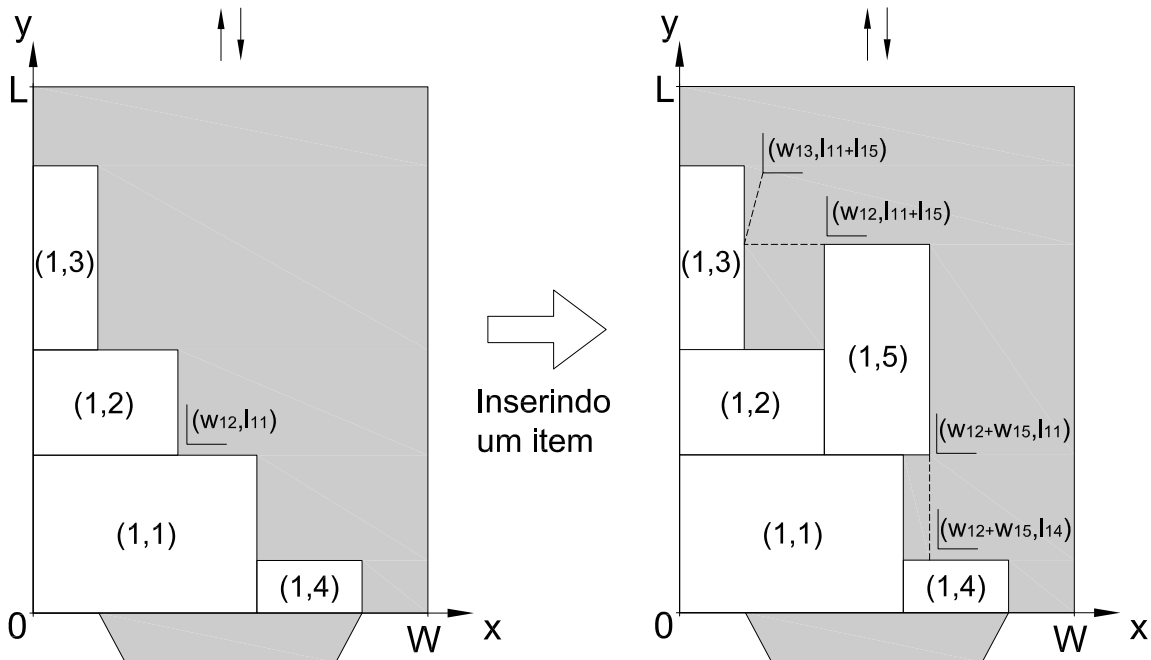


Figura 4.1: Inserindo um item no compartimento do veículo.

outro lado, as posições restantes se referem às projeções das duas inseridas anteriormente, ou seja, uma delas é a projeção mais abaixo do ponto inferior direito,  $(w_{12} + w_{15}, l_{14})$ , e a outra é a projeção mais à esquerda do ponto superior esquerdo,  $(w_{13}, l_{11} + l_{15})$ . Após criadas as novas posições, a *posList* é atualizada:  $posList = (posList - \{(w_{12}, l_{11})\}) \cup \{(w_{12} + w_{15}, l_{11}), (w_{12}, l_{11} + l_{15}), (w_{12} + w_{15}, l_{14}), (w_{13}, l_{11} + l_{15})\}$ . Vale observar que as posições duplicadas são eliminadas da lista.

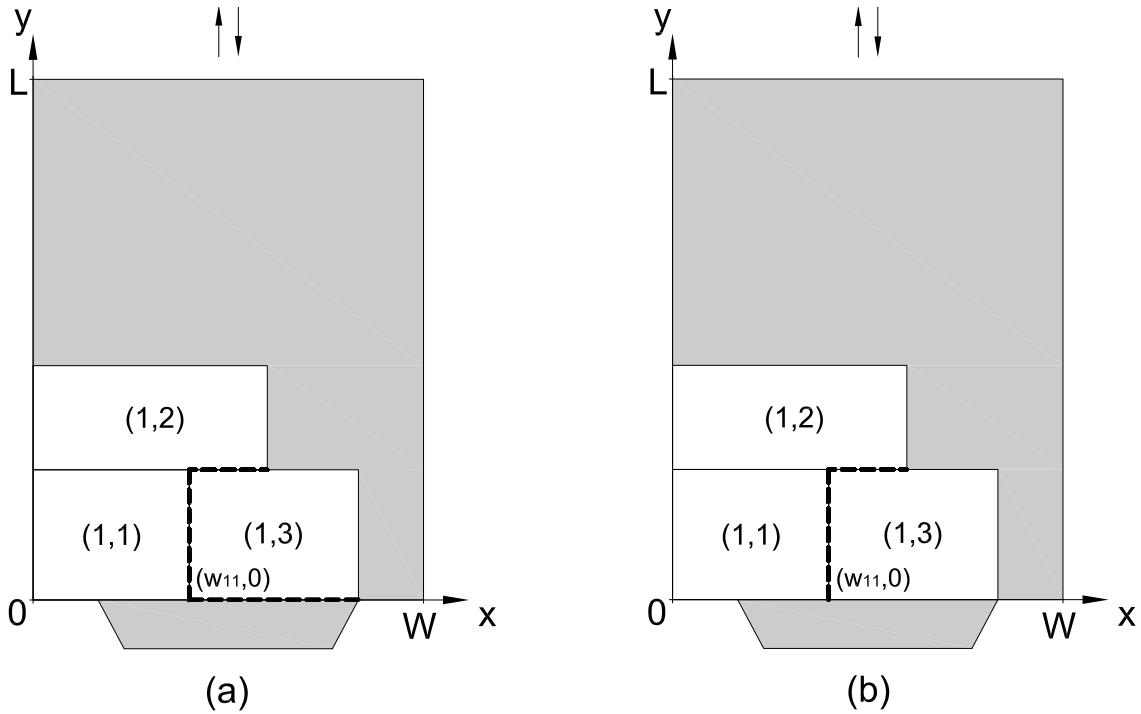
A posição onde um item deve ser inserido é determinada pela heurística utilizada. Ela deve ser viável, ou seja, não resultar em sobreposição de itens (válido para a versão sequencial e irrestrita), ou violação da ordem de entrega (para o caso sequencial). A seguir são apresentadas as cinco heurísticas construtivas utilizadas, cada uma delas é representada por  $H_i$ , com  $i = (1, \dots, 5)$ :

$H_1$ : *Bottom-Left Fill* no eixo  $W$  (Chazelle [1983]). Dentre as posições viáveis disponíveis na *posList*, a posição selecionada é aquela com a menor coordenada no eixo  $W$ . No caso de empate, é selecionada aquela posição com menor coordenada  $L$ . A tendência desta heurística é crescer como torres paralelas ao eixo  $L$ .

$H_2$ : *Bottom-Left Fill* no eixo  $L$  (Chazelle [1983]). Dentre as posições viáveis disponíveis na *posList*, a posição selecionada é aquela com a menor coordenada no eixo  $L$ . No caso de empate, é selecionada aquela posição com menor coordenada  $W$ . A tendência desta heurística é crescer como torres paralelas ao eixo  $W$ .

$H_3$ : *Max Touching Perimeter* (Lodi et al. [1999]). Para cada posição viável disponível na *posList*, o perímetro tocante do item inserido é calculado. O perímetro tocante de um item é definido como a soma das partes de suas laterais que estão em contato com outros itens já inseridos ou com as laterais do compartimento do veículo. Na Figura 4.2(a) pode-se observar o perímetro tocante do item (1,3) destacado em negrito que, no caso, é igual a  $w_{13} + l_{13} + (w_{12} - w_{11})$ . O item é colocado na posição que maximiza o perímetro tocante. A tendência desta heurística é espalhar, inicialmente, os itens nos cantos do compartimento e, posteriormente, preencher as partes mais internas.

$H_4$ : *Max Touching Perimeter No Walls* (Lodi et al. [1999]). Para cada posição viável disponível na *posList*, o perímetro tocante do item inserido é calculado. O perímetro tocante de um item é definido como a soma das partes de suas laterais que estão em contato com outros itens já inseridos. Neste caso, desconsidera-se as laterais do compartimento do veículo. Na Figura 4.2(b) pode-se observar o perímetro tocante do item (1,3) destacado em negrito que, no caso, é igual a  $l_{13} + (w_{12} - w_{11})$ . O item é colocado na posição que maximiza o perímetro tocante. Inicialmente, a tendência desta heurística é preencher a parte interna do compartimento e somente depois preencher as laterais.



**Figura 4.2:** Cálculo do perímetro tocante para as heurísticas (a) *Max Touching Perimeter* e (b) *Max Touching Perimeter No Walls*.

$H_5$ : *Min Area* (Zachariadis et al. [2009]). Para cada posição viável disponível na *posList*, a área correspondente a sua superfície retangular é calculada, como demonstrado na Figura 4.3. As áreas das superfícies retangulares referentes aos pontos  $(w_{11}, 0)$ ,  $(w_{12}, l_{11})$  e  $(0, l_{11} + l_{12})$  são iguais a  $(W - w_{11}) \times l_{11}$ ,  $(W - w_{12}) \times (L - l_{11})$  e  $W \times (L - l_{11} - l_{12})$ , respectivamente. A posição selecionada é aquela com a menor área de superfície retangular. Esta heurística visa conseguir uma melhor utilização do compartimento do veículo.

Tais heurísticas são reunidas em um só algoritmo para verificar a viabilidade do carregamento. A verificação começa pelas heurísticas mais simples ( $H_1$  e  $H_2$ ), que requerem menor esforço computacional. Caso não seja possível determinar a viabilidade, as heurísticas mais complexas são executadas (respectivamente,  $H_3$ ,  $H_4$  e  $H_5$ ). Este processo é repetido para as três ordenações disponíveis, e a qualquer momento que a viabilidade for atestada, o algoritmo termina. Caso contrário, se todas as combinações de heurísticas e ordenações se esgotarem, o algoritmo considera o carregamento inviável. O Algoritmo 1 exibe o pseudocódigo dessa coleção de heurísticas, onde *ITs* é o conjunto de itens da rota analisada, *tc* é um parâmetro que indica qual o tipo de carregamento do problema, e *Ord*<sub>1</sub>, *Ord*<sub>2</sub> e *Ord*<sub>3</sub> são as listas que armazenam os itens ordenados. Na linha 1 é verificado qual o tipo de carregamento associado ao problema em questão, se for irrestrito, as ordenações são utilizadas na ordem definida na linha 2, caso contrário, assume-se a ordem da linha 4. As variáveis auxiliares são inicializadas nas linhas 5 e 7. O laço das linhas 6-13

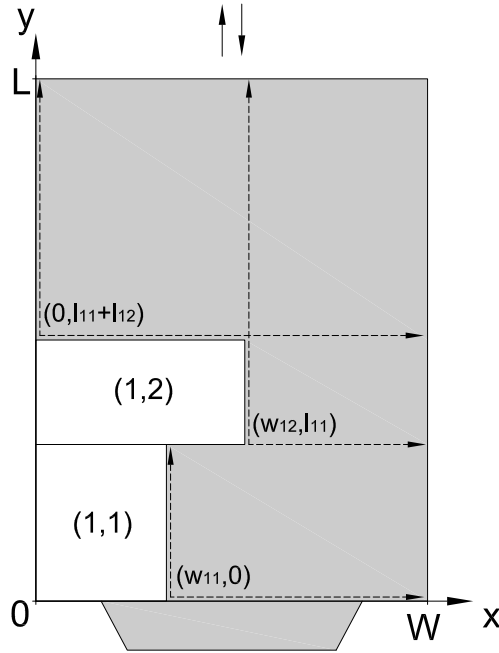


Figura 4.3: Cálculo das superfícies retangulares de cada posição disponível na *posList*.

é responsável por determinar qual a ordenação de itens é utilizada. Por outro lado, o laço das linhas 8-12 itera pelas heurísticas construtivas utilizadas para tentar atestar a viabilidade do carregamento. Na linha 9 é verificado se a heurística corrente é capaz de comprovar a viabilidade do carregamento, segundo a respectiva ordenação dos itens. Em caso positivo, retorna-se a resposta que o carregamento é viável, e em caso negativo, a heurística utilizada é alterada na linha 12. Por fim, se nenhuma combinação heurística

---

**Algoritmo 1:** Coleção de heurísticas para o carregamento.

---

**Entrada:**  $ITs$ ,  $tc$ .

**Saída:** Situação do carregamento (viável ou inviável).

```

1 se  $tc = irrestrito$  então
2   |  $Ord_1 \leftarrow ord_{unr}(ITs)$ ,  $Ord_2 \leftarrow ord_{seq}(ITs)$ ,  $Ord_3 \leftarrow ord_{Rseq}(ITs)$ ;
3 senão
4   |  $Ord_1 \leftarrow ord_{seq}(ITs)$ ,  $Ord_2 \leftarrow ord_{Rseq}(ITs)$ ,  $Ord_3 \leftarrow ord_{unr}(ITs)$ ;
5  $ordInd \leftarrow 1$ ;
6 enquanto  $ordInd \leq 3$  faça
7   |  $heuInd \leftarrow 1$ ;
8   | enquanto  $heuInd \leq 5$  faça
9     | se  $H_{heuInd}$  prova a viabilidade de  $Ord_{ordInd}$  então
10    |   | retorna O carregamento é viável;
11    |   | senão
12    |   |   |  $heuInd \leftarrow heuInd + 1$ ;
13    |   |  $ordInd \leftarrow ordInd + 1$ ;
14 retorna O carregamento é inviável;
```

---

× ordenação for suficiente para obter uma resposta positiva, o algoritmo retorna que o carregamento é inviável na linha 14.

No que diz respeito à implementação das heurísticas construtivas, a estrutura de dados utilizada para representar o compartimento do veículo é uma matriz de inteiros. Nessa matriz, as posições vazias são representadas por zeros, enquanto a fronteira de cada item inserido é identificada pelo número do cliente associado a ele. Por outro lado, a *posList* é mantida em uma estrutura de conjunto, para não haver repetições. No caso sequencial, cada posição armazenada na *posList* mantém a informação sobre quais itens ela pode conter, de modo a garantir a satisfação da restrição de carregamento sequencial. Dado que as instâncias avaliadas neste trabalho consideram apenas veículos com compartimentos de dimensões relativamente pequenas, a utilização dessas estruturas de dados apresentam um bom desempenho. Além disso, a simplicidade como o compartimento e itens são representados facilita a utilização da mesma estrutura para várias heurísticas com critérios diferentes, que é o caso.

## 4.2 *Branch-and-Bound*

Para ter a certeza se um carregamento é viável ou não, faz-se necessária a utilização de um método exato. Nesta dissertação é usado um algoritmo **BB** para resolver o carregamento dos itens em um único compartimento. Esse algoritmo foi proposto por [Martello & Vigo \[1998\]](#), como parte de um arcabouço responsável por resolver o **2BPP**. Apesar de ter sido desenvolvido para o caso de carregamentos irrestritos, [Iori et al. \[2007\]](#) propuseram uma adaptação desse método para a resolução do caso sequencial.

Antes de apresentar o método, é necessário apresentar um limite inferior para o **2BPP** proposto por [Martello & Vigo \[1998\]](#), utilizado nessa implementação. Esse limite consiste na generalização dos limites propostos por [Martello & Toth \[1990\]](#) para o Problema de *Bin Packing* Unidimensional (**1BPP**, do inglês *One-dimensional Bin Packing Problem*) e por [Dell’Amico & Martello \[1995\]](#) para o  $P||C_{max}$  (um problema de sequenciamento fortemente relacionado ao **1BPP**). Seja  $J^W = \{j \in J : w_j > W/2\}$  um conjunto de itens, o qual dois itens de  $J^W$  não podem ser empacotados lado a lado em um compartimento. Dado qualquer inteiro  $p$ , com  $1 \leq p \leq L/2$ , considera-se

$$J_1 = \{j \in J^W : l_j > L - p\}, \quad (4.1)$$

$$J_2 = \{j \in J^W : L - p \geq l_j > L/2\}, \quad (4.2)$$

Pode-se observar que dois itens de  $J_1 \cup J_2$  não podem ser empacotados no mesmo compartimento, dessa forma  $|J_1 \cup J_2|$  representa um limite inferior válido (independentemente



de  $p$ ) para a solução ótima do problema. Seja

$$J_3 = \{j \in J^W : L/2 \geq l_j \geq p\}. \quad (4.3)$$

O limite inferior pode ser fortalecido a partir da observação que nenhum item de  $J_3$  cabe em um compartimento utilizado por um item de  $J_1$ . Portanto, dado um inteiro  $p$  qualquer, tal que  $1 \leq p \leq L/2$ , um limite inferior válido pode ser dado por:

$$\Gamma_1^W(p) = \max\{\Gamma_\alpha^W(p), \Gamma_\beta^W(p)\}, \text{ onde} \quad (4.4)$$

$$\Gamma_\alpha^W(p) = |J_1 \cup J_2| + \max\left\{0, \left\lceil \frac{\sum_{j \in J_3} l_j - (|J_2|L - \sum_{j \in J_2} l_j)}{L} \right\rceil \right\}, \quad (4.5)$$

$$\Gamma_\beta^W(p) = |J_1 \cup J_2| + \max\left\{0, \left\lceil \frac{|J_3| - \sum_{j \in J_2} \left\lceil \frac{L-l_j}{p} \right\rceil}{\left\lfloor \frac{L}{p} \right\rfloor} \right\rceil \right\}. \quad (4.6)$$

De modo geral, um limite inferior válido para a solução ótima do problema é expresso por:

$$\Gamma_1^W = \max_{1 \leq p \leq L/2} \{\Gamma_1^W(p)\}. \quad (4.7)$$

Por outro lado, seja  $J^L = \{j \in J : l_j > L/2\}$ . Pode-se observar claramente que os resultados apresentados acima produzem o seguinte limite inferior de forma análoga.

$$\Gamma_1^L = \max_{1 \leq p \leq W/2} \{\Gamma_\alpha^L(p), \Gamma_\beta^L(p)\}, \quad (4.8)$$

onde  $\Gamma_\alpha^L(p)$  e  $\Gamma_\beta^L(p)$  são obtidos a partir de 4.1-4.7 por meio da substituição de  $J^W$  por  $J^L$ ,  $l_j$  por  $w_j$ , e  $L$  por  $W$ . Portanto, o limite inferior final pode ser computado como

$$\Gamma_1 = \max\{\Gamma_1^W, \Gamma_1^L\}. \quad (4.9)$$

O BB inicia seu processo fazendo o cálculo de limites inferiores (3.7 e 4.9) na raiz de sua árvore de busca, a fim de tentar evitar o esquema de enumeração. Esses limites desconsideram as condições que definem a ordem dos itens (no caso sequencial). Se o maior limite inferior for maior que 1, quer dizer que é necessário mais de um compartimento para armazenar os itens desejados, logo, o carregamento é considerado inviável. Caso contrário, os itens são ordenados em ordem não crescente de largura, desfazendo os empates segundo a ordem não crescente de comprimento, e o processo de enumeração se inicia.

A enumeração implícita da árvore de decisão do **BB** segue a ordem na qual os itens foram previamente organizados. Inicialmente, o compartimento do veículo se encontra vazio e a única posição disponível é o ponto  $(0, 0)$ . A cada nível da árvore de decisão, um item é posicionado em uma posição viável do compartimento. Dada uma solução parcial para este problema, nós descendentes da árvore são gerados para cada item ainda não inserido e para cada posição viável disponível.

O conjunto de posições viáveis disponíveis pode ser limitado a um conjunto finito de pontos associados ao contorno dos itens já posicionados. Esses pontos são denominados *corner points* por [Martello et al. \[2000\]](#), e são definidos como um conjunto  $N = \bigcup_{i \in V \setminus \{0\}} N(i)$ , tal que  $N(i) = \{(x, y) : \forall t \in \hat{m}_i, x \geq x_{it} + w_{it} \text{ ou } y \geq y_{it} + l_{it}\}$ , e  $\hat{m}_i$  representa todos os itens já inseridos do cliente  $i$ . Ou seja,  $N$  é o conjunto de posições nas quais os itens a serem inseridos não podem se mover em direção ao eixo  $y = 0$  (para baixo) e nem em direção ao eixo  $x = 0$  (para a esquerda). Contudo, no caso sequencial, novas posições podem ser criadas devido ao fato de um item poder bloquear a retirada de itens de um cliente que será atendido antes.

A Figura 4.4 ilustra um compartimento de um veículo com um carregamento parcial. Esse carregamento é composto de cinco itens dos clientes associados a uma rota  $(S, \sigma)$ , onde  $S = \{1, 2, 3\}$  e  $\sigma(1) = 1$ ,  $\sigma(2) = 2$  e  $\sigma(3) = 3$ . A linha destacada em negrito indica o contorno dos itens, e os pontos  $(0, l_{21} + l_{11})$ ,  $(w_{11}, l_{32} + l_{31})$  e  $(w_{21} + w_{22} + w_{31}, l_{32})$  são os *corner points*, indicados pelos círculos preenchidos. Segundo [Scheithauer \[1997\]](#), a área hachurada, sobre o item  $(2, 2)$ , não precisa ser explorada, o que resulta em uma redução no tamanho da árvore de enumeração. Considerando o carregamento com a restrição sequencial, um novo item do cliente 1 poderia ser posicionado em qualquer *corner point*, enquanto somente as posições  $(w_{11}, l_{32} + l_{31})$  e  $(w_{21} + w_{22} + w_{31}, l_{32})$  estariam disponíveis para o cliente 2. Por outro lado, para o cliente 3, a princípio, só haveria a possibilidade de posicionar um novo item no ponto  $(w_{21} + w_{22} + w_{31}, l_{32})$ , sem que houvesse alguma obstrução. Contudo, quando há a possibilidade de um item já posicionado ter o seu caminho bloqueado, uma nova posição pode ser criada. Na Figura 4.4, isso acontece com o item  $(2, 2)$ , que pode ser bloqueado por um item do cliente 3 e, portanto, uma posição adicional é criada no ponto  $(w_{21} + w_{22}, l_{32} + l_{31})$  (representada por um círculo vazio). De modo geral, as posições adicionais são criadas entre dois *corner points* consecutivos, exatamente onde há divisões de regiões onde alguns itens podem ser colocados e outros não.

Para acelerar o algoritmo, um limite inferior é calculado sempre que um novo item é inserido no compartimento do veículo, na tentativa de executar uma poda na árvore do **BB**. Este limite foi usado por [Martello et al. \[2000\]](#) para o **2BPP** com um único compartimento. O objetivo é medir a área total desperdiçada (ou seja, espaços abaixo do contorno que não há itens) por um carregamento parcial e compará-la com o maior desperdício que já houve até então. Caso o primeiro seja maior que o último, o nó da

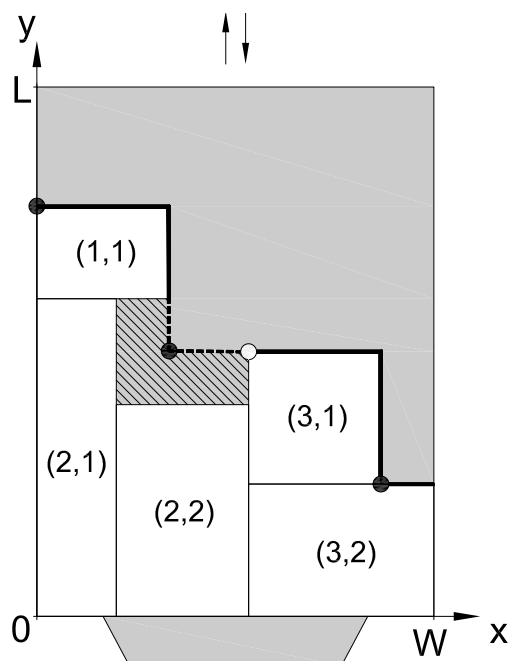


Figura 4.4: Carregamento parcial com a linha de contorno e os *corner points* destacados.

árvore é podado, pois ele não levaria a uma solução melhor que as obtidas anteriormente.

Outras verificações são feitas para tentar podar mais nós da árvore de enumeração. Para o caso do carregamento irrestrito, é verificada se a área disponível acima do contorno dos itens é maior ou igual a área total dos itens ainda não inseridos. Em caso negativo, a poda é executada. Com relação ao caso sequencial, duas comparações são realizadas para cada cliente: (a) considera-se as subáreas disponíveis para os itens a serem inseridos, e (b) considera-se as larguras das subáreas disponíveis e as compara com o item mais largo a ser inserido. Caso os itens de algum cliente não caibam nos espaços disponíveis ou algum item seja mais largo que o espaço destinado a ele, o nó deve ser podado.

Diferentemente da implementação utilizada para as heurísticas da Seção anterior, no **BB** o compartimento dos veículos é representado pela mesma estrutura de dados adotada por [Burke et al. \[2004\]](#). Isto é, um vetor com o número de elementos igual à largura do compartimento. Cada elemento desse vetor corresponde ao comprimento total já armazenado em cada coordenada da largura do compartimento. Pode-se observar que os elementos do vetor formam um contorno do carregamento do veículo, o que facilita a identificação dos *corner points* pelo algoritmo. Por sua vez, os *corner points* são armazenados em um conjunto, para evitar repetições (assim como a implementação da *posList* na Seção anterior). Pela eficiência de como são feitas a inserção e a remoção de itens, além da identificação de *corner points*, a utilização dessas estruturas de dados se encaixam bem no arcabouço de um algoritmo exato como o **BB**, onde aspectos como memória utilizada e tempo de execução são críticos. As estruturas aqui descritas também são válidas para o algoritmo da Seção a seguir.

### 4.3 *Branch-and-Bound* Aproximado

Por muitas vezes o algoritmo **BB** exige um alto tempo computacional para obter uma resposta, principalmente quando o carregamento é inviável. Por isso, algumas estratégias são aplicadas para limitar o espaço de busca do **BB** visando reduzir de maneira significativa o seu tempo de execução. Ibaraki [1987] apresenta algumas técnicas de fácil implementação que são frequentemente empregadas. Dentre elas, está o método de enumeração  $\mu$ -cut, onde em cada nó da árvore de **BB** apenas os  $\mu \in \mathbb{N}$  primeiros nós filhos são considerados.

Nesta dissertação foi desenvolvida uma variação do método  $\mu$ -cut, a qual foi denominada  $\mu$ -cut Decrescente. Nessa variante, inicialmente, todos os itens podem ser posicionados na posição  $(0, 0)$ . Após inserido o primeiro item, a limitação característica do  $\mu$ -cut é aplicada. Contudo, neste caso, o valor de  $\mu$  é decrescido de uma unidade a cada nível da árvore de enumeração, de tal forma que o valor mínimo que  $\mu$  pode assumir é 2. Assim que uma disposição viável com todos os itens for encontrada, o algoritmo retorna o resultado atestando a viabilidade, caso contrário, o carregamento é considerado inviável.

### 4.4 Acelerando a Verificação com uma Árvore de Viabilidade

Devido ao tempo gasto com as verificações dos carregamentos, é interessante manter uma estrutura que armazene informações sobre a viabilidade daqueles que já foram testados. Para esse fim, uma árvore de viabilidade foi criada para armazenar as rotas associadas aos carregamentos avaliados. Apesar de estruturas similares terem sido usadas nos trabalhos da literatura que abordam o **2L-CVRP**, é a primeira vez que uma estrutura desse tipo é usada para inferir a viabilidade de carregamentos ainda não analisados (conforme é mostrado nas Subseções a seguir).

A raiz da árvore de viabilidade representa o depósito, os outros nós representam os clientes e o caminho desde a raiz até um nó qualquer representa uma rota do **2L-CVRP** (define-se que após o último nó considerado o caminho continua até o depósito). Além disso, cada nó da árvore contém a informação sobre a viabilidade da rota no cliente que ele representa, isso é associado ao nó por meio de um rótulo *stt*, que assume o valor 1 se o carregamento da rota for viável, 2 se for inviável e 0 se ainda for indeterminada. Esse último caso ocorre quando não se pode afirmar nada sobre a viabilidade dos trechos de uma rota inserida. O modo como a árvore de viabilidade é populada e como é feita a recuperação da informação contida nela são explicados nas Subseções 4.4.1 e 4.4.2, respectivamente.

### 4.4.1 Inserção na Árvore

Quando a informação sobre uma rota não é encontrada (ou é indeterminada) na árvore, a sua viabilidade é avaliada por um dos métodos de verificação e o resultado é inserido na estrutura. O modo como as rotas são inseridas na árvore de viabilidade diverge quanto ao tipo de carregamento (sequencial ou irrestrito). No caso sequencial, a ordem em que os clientes são visitados interfere na viabilidade do carregamento, portanto, essa ordem deve ser respeitada no momento da inserção na árvore. Por outro lado, no caso irrestrito não há esta limitação, desta forma, para evitar informações replicadas na árvore, os clientes são ordenados crescentemente segundo seus índices antes de ser feita a inserção.

As inserções de rotas com carregamentos viáveis e inviáveis também se diferem. Porém, para que elas sejam explicadas, é necessário a consideração de duas propriedades:

**Propriedade 1.** *Dada uma rota  $(S, \sigma)$  com o carregamento viável, toda rota  $(S', \sigma')$ , subsequência de  $(S, \sigma)$ , também detém um carregamento viável.*

*Demonstração.* Considere o fato que todo cliente tem uma demanda não negativa de itens. Portanto, ao ser retirado um cliente  $i \in S \setminus S'$  de  $S$ , pelo menos um item será retirado do carregamento de  $S$ . Dado que um carregamento viável é caracterizado por um ou mais itens posicionados, sem sobreposições, no mesmo compartimento, ao se retirar os itens do cliente  $i$ , o carregamento continuará viável, uma vez que os itens remanescentes permanecerão intactos. Consequentemente, este passo pode ser repetido para todos os clientes em  $S \setminus S'$ , de tal forma a sobrar somente os itens dos clientes em  $S'$ . Portanto, a viabilidade de  $(S', \sigma')$  é garantida.  $\square$

**Propriedade 2.** *Dada uma rota  $(S', \sigma')$  com o carregamento inviável, toda rota  $(S, \sigma)$ , tal que  $(S', \sigma')$  é uma sub-rota de  $(S, \sigma)$ , também detém um carregamento inviável.*

*Demonstração.* É fácil observar que a contrapositiva desta afirmação é exatamente igual à Propriedade 1, que já foi provada verdadeira. Logo, esta Propriedade também é.  $\square$

A Figura 4.5 ilustra todas as possibilidades de inserção existentes na árvore de viabilidade. Inicialmente a árvore se encontra vazia, com apenas o nó raiz. Logo em seguida é mostrada a inserção da rota  $(S^1, \sigma^1) = (0 - 1 - 6 - 5 - 3 - 0)$ , que tem carregamento inviável. Neste caso, somente o último nó é marcado como inviável, enquanto os outros nós, referentes a esta rota, são marcados como indeterminados, pois não há garantia que se for retirado o último cliente a rota continuará inviável ou não. Uma segunda inserção adiciona a rota com carregamento viável  $(S^2, \sigma^2) = (0 - 1 - 3 - 2 - 0)$ . Pode-se observar que ao ser inserida, todos os nós referentes a esta rota são marcados como viáveis, o que é válido segundo a Propriedade 1. Finalmente, é feita mais uma inserção de uma rota com carregamento inviável,  $(S^3, \sigma^3) = (0 - 1 - 6 - 5 - 0)$ . Este caso, em especial, resulta em uma poda na árvore de viabilidade, pois, segundo um caso específico da Propriedade 2,



Dada a árvore resultante das inserções na Figura 4.5, alguns exemplos dos casos acima podem ser analisados, considerando o caso sequencial. As rotas (0 - 1 - 3 - 0), (0 - 2 - 3 - 1 - 0) e (0 - 3 - 1 - 0) são rotas com carregamentos viáveis, sendo a primeira verificada diretamente e as outras duas verificadas após a inversão na ordem de visitação. Por outro lado, as rotas (0 - 3 - 1 - 6 - 5 - 2 - 0) e (0 - 7 - 5 - 6 - 1 - 0) possuem carregamento inviável, ambas constatadas pela segunda etapa de verificação direta e inversa, respectivamente. Finalmente, dois casos onde há carregamentos indeterminados são observados em (0 - 1 - 5 - 0) e (0 - 1 - 2 - 3 - 0), sendo o primeiro definido na árvore como indeterminado, e o segundo, não encontrado.





# Capítulo 5

## Métodos para Resolução do 2L-CVRP

Por se tratar de um problema muito complexo e de difícil resolução, na grande maioria das vezes, os métodos heurísticos são escolhidos para a resolução do 2L-CVRP (conforme exibido na Seção 2.3). Até onde se sabe, nenhuma heurística baseada em programação matemática foi desenvolvida até então. Motivado por isso, duas heurísticas baseadas em Geração de Colunas (CG) foram desenvolvidas neste trabalho. Tais abordagens são apresentadas na Seção 5.1.

Vale observar, segundo a Tabela 2.1, que após o trabalho de Iori et al. [2007], a maioria dos métodos desenvolvidos não seguiram as mesmas características propostas pelo autor (ou seja, número de rotas igual ao número de veículos, número de clientes por rota maior que um, e distâncias truncadas entre clientes). Por consequência, somente para essas características que existem soluções ótimas conhecidas. Devido a esse fato, nesta dissertação foi implementado um BC baseado naquele de Iori et al. [2007] com a permissão de rotas com somente um cliente e o número de rotas menor ou igual ao número de veículos, além de considerar as distâncias reais entre os clientes. As modificações no modelo e as características do algoritmo são apresentadas na Seção 5.2.

### 5.1 Heurísticas Baseadas em Geração de Colunas

O método CG é utilizado para resolver problemas de programação linear com um número muito grande de variáveis, contudo pode também ser usado para obter um limite dual para um problema de programação inteira com a mesma característica. No caso do 2L-CVRP, o algoritmo CG é usado para obter limites inferiores por meio do modelo da Seção 3.2, denominado Problema Mestre (MP, do inglês *Master Problem*) do método. Esses limites são obtidos através da relaxação linear do MP, nomeado Problema Mestre Linear (LMP, do inglês *Linear Master Problem*), por meio da substituição da restrição

3.11 pela desigualdade:

$$0 \leq \lambda_r \leq 1 \quad \forall r \in R, \quad (5.1)$$

a qual, devido às restrições 3.9, pode ser simplificada à seguinte desigualdade:

$$\lambda_r \geq 0 \quad \forall r \in R. \quad (5.2)$$

A cada iteração do método simplex as variáveis não-básicas são precificadas e entram na base caso tenham custo reduzido negativo. Ou seja, dado os vetores das variáveis duais livres  $\pi$  e  $\tau$ , associados às restrições 3.9 e 3.10, respectivamente, deseja-se encontrar uma coluna  $r \in R$  tal que:

$$\min_{r \in R} \bar{c}_r := c_r - \sum_{i \in V \setminus \{0\}} \alpha_{ir} \pi_i - \tau. \quad (5.3)$$

A precificação explícita de todas as variáveis é uma operação muito custosa quando  $|R|$  é muito grande. Portanto, a ideia é trabalhar com um pequeno subconjunto  $R' \subseteq R$  ( $|R'| \ll |R|$ ), dando origem ao Problema Mestre Linear Restrito (**RLMP**, do inglês *Restricted Linear Master Problem*), onde a avaliação dos custos reduzidos é feita por meio de enumeração implícita.

Seja  $\bar{\lambda}$  uma solução básica para o **RLMP, e  $\bar{\pi}$  e  $\bar{\tau}$  as suas respectivas variáveis duais. Dado que o custo  $c_r$  de uma coluna  $r \in R$  pode ser calculado a partir de uma função  $c$ , tem-se que o subproblema**

$$\bar{c}^* := \min_{r \in R} \left\{ c(r) - \sum_{i \in V \setminus \{0\}} \alpha_{ir} \bar{\pi}_i - \bar{\tau} \right\} \quad (5.4)$$

realiza a precificação. Se  $\bar{c}^* \geq 0$ , então significa que não existe nenhum custo reduzido  $\bar{c}_r$  com valor negativo e a solução do **RLMP** não pode ser melhorada, ou seja,  $\bar{\lambda}$  representa a solução ótima para o **MP**. Caso contrário, a coluna com custo reduzido igual a  $\bar{c}^*$  é adicionada ao **RLMP** e o processo é então repetido até que nenhuma coluna com custo reduzido negativo seja encontrada. Inicialmente, esse processo deve ser inicializado com uma solução viável para o **RLMP** — tal solução pode ser artificial ou então obtida por algum outro método — para garantir que as variáveis duais passadas para o subproblema de precificação sejam válidas. O Algoritmo 2 exibe o pseudocódigo de um método de **CG** genérico. Na linha 1 o *pool* inicial de colunas é armazenado na lista *cols*. No laço das linhas 2-6, retrata-se o processo responsável pela resolução do Problema Linear e da adição de novas colunas. As colunas são adicionadas ao **RLMP** na linha 3, o qual é resolvido à otimalidade na linha 4. Em seguida, a lista de colunas é esvaziada e possíveis

novas colunas são adicionadas a mesma, respectivamente, nas linhas 5 e 6. O laço das linhas 2-6 é executado enquanto houverem colunas a serem inseridas no modelo. Ao final do processo, a solução ótima do RLMP é retornada na linha 7. Os detalhes sobre a geração das soluções iniciais, e o subproblema de precificação e seus métodos de resolução podem ser encontrados nas Subseções 5.1.1 e 5.1.2, respectivamente.

---

**Algoritmo 2:** Método *Geração de Colunas*

---

**Entrada:** RLMP.

**Saída:** Solução ótima do RLMP.

```

1  $cols \leftarrow$  Inicialize o pool de colunas;
2 enquanto  $|cols| > 0$  faça
3   Adicione  $cols$  ao RLMP;
4    $sol \leftarrow$  Resolva(RLMP);
5    $cols \leftarrow \emptyset$ ;
6    $cols \leftarrow$  Gere novas colunas;
7 retorna  $sol$ ;

```

---

As heurísticas baseadas em CG são desenvolvidas quando o método exato exige um tempo computacional muito alto, especialmente por conta do subproblema de precificação. Desse modo, alguns artifícios são utilizados ou para acelerar a convergência do método, ou para tirar proveito das colunas geradas durante um período de tempo. Nas Subseções 5.1.3 e 5.1.4 são apresentadas duas heurísticas baseadas em CG e suas características.

### 5.1.1 Geração do *Pool* Inicial de Colunas

As colunas do *pool* inicial são geradas por várias heurísticas diferentes. O objetivo destas é criar soluções para o 2L-CVRP, as quais, uma vez que são compostas de rotas viáveis, representam um conjunto de colunas válidas para o modelo de SP. No início, heurísticas construtivas são utilizadas para gerar as soluções, posteriormente, perturbações e buscas locais podem ser aplicadas para diversificar e melhorar as soluções, respectivamente. As heurísticas construtivas e as vizinhanças utilizadas neste trabalho fazem uso apenas de soluções viáveis, para isso, toda mudança na solução resulta em refazer a verificação nos carregamentos dos veículos.

Com relação às heurísticas construtivas, são utilizados quatro tipos que se diferenciam quanto à inserção dos clientes nas rotas. Apesar disso, essas heurísticas têm um objetivo em comum: construir soluções com o número de rotas igual ao número de veículos. A seguir são detalhadas suas características:

**Nearest Customer.** Dada uma rota  $(S, \sigma)$ , a cada iteração são avaliados todos os clientes  $i \in V \setminus \{0\}$  ainda não roteados. Dentre esses, aquele que possui a menor

distância  $(i, j)$ , tal que  $j \in S$ , é inserido na rota na posição com menor incremento no valor da solução. Caso o veículo não suporte a inserção do cliente escolhido, uma nova rota é criada e o processo continua até que todos os clientes estejam alocados.

**Nearest Neighbor.** Dada uma rota  $(S, \sigma)$ , a cada iteração são avaliados todos os clientes  $i \in V \setminus \{0\}$  ainda não roteados. Dentre esses, aquele que possui a menor distância  $(i, j)$ , tal que  $\sigma(j) = |S|$ , é inserido na última posição da rota. Caso o veículo não suporte a inserção do cliente escolhido, uma nova rota é criada e o processo continua até que todos os clientes estejam alocados.

**Smallest Long Edge.** Dada uma rota  $(S, \sigma)$ , a cada iteração são avaliados todos os clientes  $i \in V \setminus \{0\}$  ainda não roteados. Para cada um desses, é selecionada a maior aresta que o conecta a qualquer  $j \in S$ . Dentre essas arestas, aquela que possui o menor valor tem o vértice  $i$  inserido na rota na posição com menor incremento no valor da solução. Caso o veículo não suporte a inserção do cliente escolhido, uma nova rota é criada e o processo continua até que todos os clientes estejam alocados.

**Best Fit Decreasing.** Seja  $i \in V \setminus \{0\}$  um cliente não roteado, selecionado aleatoriamente. Esse cliente deve ser inserido na rota a qual o espaço vazio no compartimento de carregamento do veículo associado seja o mínimo possível e, ao mesmo tempo, caiba os seus itens. Além disso, a posição na rota onde o cliente deve ser inserido é aquela com menor incremento no valor da solução. Neste procedimento, o número de rotas inicia-se igual ao número de veículos.

Ao contrário das heurísticas construtivas, os procedimentos de perturbação e busca local são utilizados para alterar soluções já existentes. Ao todo, quatro estruturas de vizinhança foram usadas para desenvolver esses procedimentos, são elas: *inversion*, *exchange*, *1-OPT* e *2-OPT*. A primeira considera somente movimentos intra-rota, a segunda e a terceira são válidas para movimentos intra e inter-rotas, e a última só realiza movimentos inter-rotas. A vizinhança *inversion* de uma solução consiste nas soluções que diferenciam da primeira por meio da inversão na ordem de visitação dos clientes em um trecho de uma rota. A vizinhança *exchange* de uma solução é definida como o conjunto de soluções que diferem da primeira através da troca de posição entre dois clientes quaisquer. A vizinhança *1-OPT* de uma solução representa as soluções que diferem da primeira devido a realocação de um cliente. Por fim, a vizinhança *2-OPT* de uma solução consiste no conjunto de soluções que diferem da primeira com relação a troca de duas arestas de rotas diferentes.

Foram desenvolvidos três mecanismos de perturbação para diversificar as soluções obtidas, isto é: perturbação *inversion*, perturbação *1-OPT* e perturbação *exchange*. Todos eles utilizam as vizinhanças de mesmo nome explicadas anteriormente. Nesse caso,

as perturbações realizam somente um movimento aleatório por vez, dentro de suas respectivas vizinhanças. Além disso, a fim de obter soluções suficientemente diferentes da solução inicial, é ideal que as perturbações diferentes sejam usadas em sequência e mais de uma vez.

Por sua vez, quatro buscas locais foram desenvolvidas para aprimorar a qualidade das soluções obtidas. Todas elas usam a estratégia melhor aprimorante para realizar seus movimentos, ou seja, para cada movimento possível, todas as possibilidades são avaliadas e a de menor custo, melhor que a solução corrente, é efetivada. Tais buscas locais são definidas como segue:

**Busca Local *Inversion*.** Esta busca explora a vizinhança *inversion*. Dada uma rota  $(S, \sigma)$  selecionada aleatoriamente e um cliente  $i \in S$  selecionado da mesma forma, a cada iteração, o custo total de inverter o trecho de rota compreendido entre todos os pares  $(i, j)$ , tal que  $j \in S \setminus \{i\}$ , são avaliados, e aquela inversão que for a melhor aprimorante é executada. A Figura 5.1 ilustra um movimento resultante de uma iteração desta busca local, onde a inversão ocorre entre o par de clientes  $(3, 4)$ .

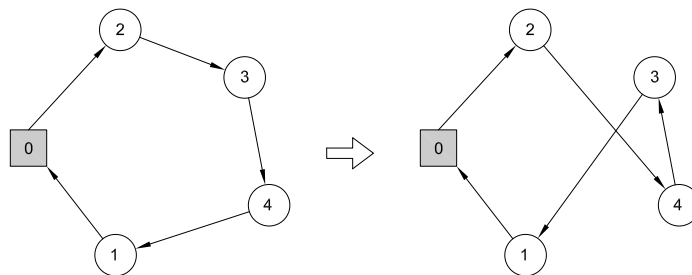


Figura 5.1: Exemplo de um movimento da busca local *Inversion*.

**Busca Local *Exchange*.** Esta busca explora a vizinhança *exchange*. Dado um cliente  $i \in V \setminus \{0\}$  selecionado aleatoriamente, a cada iteração, o custo total de efetuar as trocas de posições entre  $i$  e  $j \in V \setminus \{0, i\}$  são avaliadas, e aquela troca que for a melhor aprimorante é executada. A Figura 5.2 ilustra um movimento resultante de uma iteração desta busca local, onde a troca ocorre entre os clientes 1 e 5, pertencentes a rotas diferentes.

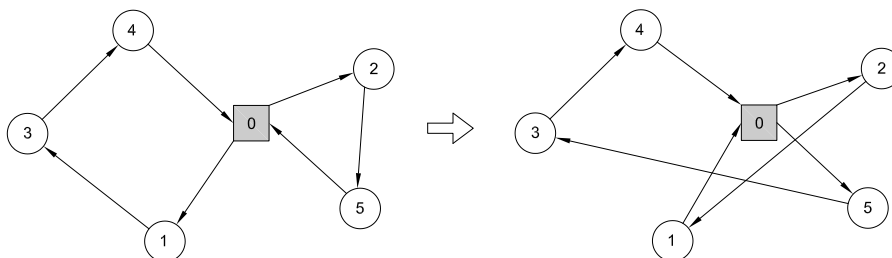
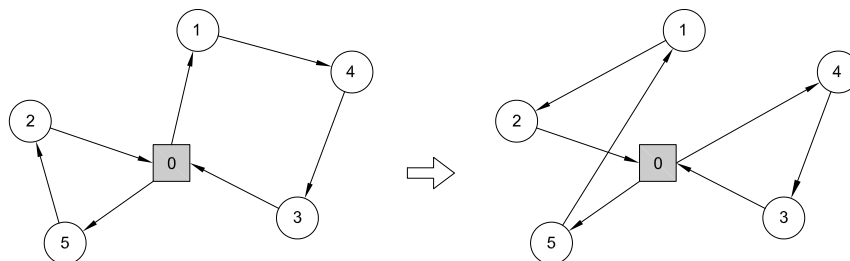


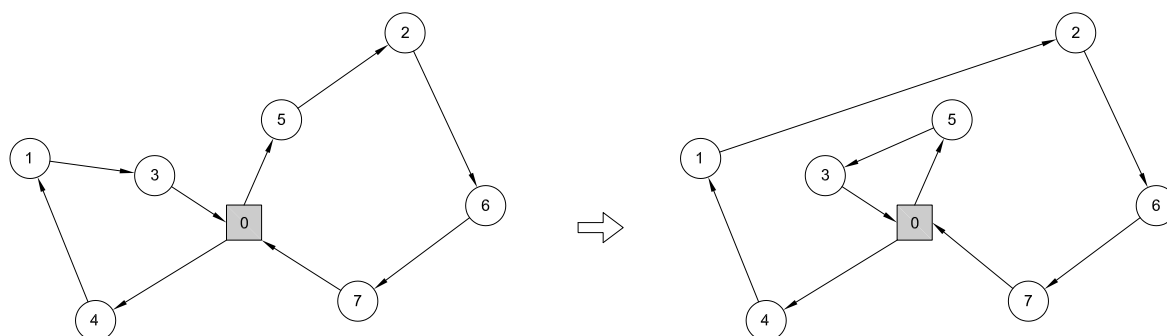
Figura 5.2: Exemplo de um movimento da busca local *Exchange*.

**Busca Local 1-*OPT*.** Esta busca explora a vizinhança 1-*OPT*. Dado um cliente  $i \in V \setminus \{0\}$  selecionado aleatoriamente, a cada iteração, o custo total de realocá-lo em todas as posições das rotas são avaliados, e aquela realocação que for a melhor aprimorante é executada. A Figura 5.3 ilustra um movimento resultante de uma iteração desta busca local, onde o cliente 1 foi realocado entre os clientes 2 e 5, pertencentes a outra rota.



**Figura 5.3:** Exemplo de um movimento da busca local 1-*OPT*.

**Busca Local 2-*OPT*.** Esta busca explora a vizinhança 2-*OPT*. Dada uma rota  $(S, \sigma)$  selecionada aleatoriamente e uma aresta  $(i, j)$ , tal que  $i, j \in S$  e  $i \neq j$ , selecionada da mesma forma, a cada iteração, o custo total das trocas dos extremos entre as arestas  $(i, j)$  e  $(i', j')$ , tal que  $i', j' \in S'$ ,  $S' \neq S$  e  $i' \neq j'$ , são avaliadas, e aquela troca que for a melhor aprimorante é executada. A Figura 5.4 ilustra um movimento resultante de uma iteração desta busca local, onde as arestas  $(1, 3)$  e  $(5, 2)$  tem as suas extremidades trocadas, tal que o cliente 5 se conecta ao cliente 3, e o cliente 1 ao 2, resultado nas novas arestas  $(5, 3)$  e  $(1, 2)$ .



**Figura 5.4:** Exemplo de um movimento da busca local 2-*OPT*.

A partir dessas estruturas, dois algoritmos baseados na metaheurística *Iterated Local Search* (ILS) (Lourenço et al. [2010]) foram desenvolvidos. Esse método se baseia na ideia que um procedimento de busca local pode ser melhorado gerando-se novas soluções de partida, as quais são obtidas por meio de perturbações numa solução possivelmente ótima local. A perturbação deve permitir que a busca local explore diferentes soluções e, além disso, deve evitar um reinício aleatório. O ILS é, portanto, um método de busca

local que procura focar a busca em um pequeno subespaço de soluções, definido por possíveis ótimos locais. Basicamente, um algoritmo baseado no **ILS** deve conter as seguintes etapas: (a) geração da solução inicial, geralmente obtida através das heurísticas construtivas, (b) busca local, responsável por aprimorar as soluções utilizadas, (c) perturbação, necessária para evitar a estagnação do algoritmo, e (d) critério de aceitação, responsável por promover a atualização da solução corrente mediante um critério.

Ambos algoritmos propostos tiveram suas estruturas definidas de acordo com resultados obtidos em testes preliminares. No caso, esses testes colaboraram para determinar quais as estruturas de perturbação e buscas locais seriam usadas, e em qual ordem. As organizações utilizadas pelos algoritmos foram decididas de acordo com aquelas que obtiveram melhores resultados. Além disso, foi definido que o critério de aceitação considerado nas adaptações feitas neste trabalho consiste em sempre atualizar a solução corrente, independentemente de regras.

A primeira heurística baseada em **ILS** desenvolvida, denominada ILS-1, é exibida em forma de pseudocódigo no Algoritmo 3. Sejam  $sol$  a solução utilizada nas operações ao decorrer do algoritmo e  $sol^*$  a melhor solução encontrada. Na linha 1,  $sol$  é inicializada com a solução resultante da heurística construtiva *Nearest Customer*. Logo em seguida, nas linhas 2-4, são aplicadas três buscas locais, respectivamente, 1-*OPT*, *Exchange* e 2-*OPT*, que executam somente uma iteração cada. Na linha 5 a solução  $sol^*$  é atualizada. O laço das linhas 6-16 define o mecanismo principal de um **ILS**, composto

---

**Algoritmo 3:** Heurística *ILS-1*


---

**Entrada:**  $CP_1, CP_2, CP_3$ .

**Saída:** A melhor solução para o **2L-CVRP** encontrada pelo algoritmo.

```

1  $sol \leftarrow$  GereSoluçãoInicial(NearestCustomer);
2  $sol \leftarrow$  BuscaLocal(1-OPT,  $sol$ , 1 iteração);
3  $sol \leftarrow$  BuscaLocal(Exchange,  $sol$ , 1 iteração);
4  $sol \leftarrow$  BuscaLocal(2-OPT,  $sol$ , 1 iteração);
5  $sol^* \leftarrow sol$ ;
6 enquanto critério de parada  $CP_1$  não for atendido faça
7   enquanto critério de parada  $CP_2$  não for atendido faça
8      $sol \leftarrow$  Perturbação(Exchange,  $sol$ );
9      $sol \leftarrow$  Perturbação(1-OPT,  $sol$ );
10  enquanto critério de parada  $CP_3$  não for atendido faça
11     $sol \leftarrow$  BuscaLocal(1-OPT,  $sol$ , completo);
12     $sol \leftarrow$  BuscaLocal(2-OPT,  $sol$ , completo);
13     $sol \leftarrow$  BuscaLocal(Inversion,  $sol$ , completo);
14     $sol \leftarrow$  BuscaLocal(Exchange,  $sol$ , completo);
15  se  $f(sol) < f(sol^*)$  então
16     $sol^* \leftarrow sol$ ;
17 retorna  $sol^*$ ;

```

---

pelas perturbações seguidas das buscas locais, e é executado até a condição de parada  $CP_1$  ser atendida. As perturbações *Exchange* e *1-OPT* (linhas 8 e 9) são executadas em sequência pelo laço das linhas 7-9, até que a condição de parada  $CP_2$  seja atendida. Em seguida, um outro laço nas linhas 10-14 executa quatro buscas locais completas, ou seja, cada busca é executada até que não seja possível mais melhoras. As buscas locais são executadas nas linhas 11-14, na ordem: *1-OPT*, *2-OPT*, *Inversion* e *Exchange*. O laço (linhas 10-14) termina quando a condição de parada  $CP_3$  for atendida. Na linha 15 é verificado se o custo de  $sol$  é menor que o de  $sol^*$  e, se for verdadeiro,  $sol^*$  é atualizada na linha 16. Finalmente, a melhor solução obtida é retornada na linha 17.

O Algoritmo 4 exibe o pseudocódigo da segunda heurística baseada em ILS desenvolvida, denominada ILS-2. Assim como no pseudocódigo do ILS-1,  $sol$  e  $sol^*$  possuem o mesmo significado. Na linha 1 é gerada uma solução inicial por meio da heurística construtiva *Nearest Neighbor*, e em seguida (linha 2) a solução  $sol^*$  é atualizada. No laço das linhas 3-11, são executadas as perturbações e buscas locais até que a condição de parada  $CP_1$  seja atendida. Inicialmente, duas perturbações, *Exchange* e *1-OPT*, são aplicadas à solução  $sol$  nas linhas 4 e 5, respectivamente. Logo depois, um laço nas linhas 6-9 executa três buscas locais de uma iteração cada. Esse laço é composto das buscas *1-OPT*, *Exchange* e *2-OPT*, e é executado até que se atenda a condição de parada  $CP_2$ . Se a solução  $sol$  tiver um custo melhor que aquele de  $sol^*$ , a última é atualizada na linha 11, após verificação na linha 10. Por fim, a melhor solução encontrada é retornada na linha 12.

---

**Algoritmo 4:** Heurística *ILS-2*


---

**Entrada:**  $CP_1, CP_2$ .

**Saída:** A melhor solução para o 2L-CVRP encontrada pelo algoritmo.

```

1  $sol \leftarrow$  GereSoluçãoInicial(NearestNeighbor);
2  $sol^* \leftarrow sol$ ;
3 enquanto critério de parada  $CP_1$  não for atendido faça
4    $sol \leftarrow$  Perturbação(Exchange,  $sol$ );
5    $sol \leftarrow$  Perturbação(1-OPT,  $sol$ );
6   enquanto critério de parada  $CP_2$  não for atendido faça
7      $sol \leftarrow$  BuscaLocal(1-OPT,  $sol$ , 1 iteração);
8      $sol \leftarrow$  BuscaLocal(Exchange,  $sol$ , 1 iteração);
9      $sol \leftarrow$  BuscaLocal(2-OPT,  $sol$ , 1 iteração);
10    se  $f(sol) < f(sol^*)$  então
11       $sol^* \leftarrow sol$ ;
12 retorna  $sol^*$ ;

```

---

O *pool* inicial de colunas é gerado pela combinação dos métodos citados nesta Subseção. Essa combinação foi elaborada com o objetivo de que o seu resultado fosse o mais diversificado possível, contudo, com pelo menos um conjunto de colunas que compusesse



uma solução de custo razoável. Para isso, as heurísticas construtivas seguidas de perturbações foram utilizadas para a obtenção de soluções variadas, e o ILS-1 e ILS-2 foram usados para garantir a existência de uma solução satisfatória. O método resultante é ilustrado em forma de pseudocódigo no Algoritmo 5, onde  $pool$  define o conjunto das colunas criadas,  $HC_1$ ,  $HC_2$ ,  $HC_3$  e  $HC_4$  se referem às heurísticas construtivas desenvolvidas,  $sol_1$ ,  $sol_2$  e  $sol$  são soluções auxiliares, e  $max\_cols$  representa o número máximo de colunas que podem ser geradas. Na linha 1, o  $pool$  é inicializado como vazio, enquanto na linha 2, as heurísticas construtivas são associadas às estruturas  $HC_i$ , com  $i \in [1, 4]$ . Duas soluções são obtidas nas linhas 3 e 4 por meio das heurísticas ILS-1 e ILS-2, respectivamente, e ambas têm as rotas inseridas no  $pool$  de colunas (linha 5). O laço das linhas 6-12 é estruturado para criar soluções diversificadas para preencher o  $pool$  até o número máximo de colunas ser alcançado. Na linha 7 é sorteado um número no intervalo de 1 a 4, referente ao índice da heurística construtiva a ser usada na linha 8 para criar uma solução  $sol$ . As três próximas linhas executam as perturbações *Exchange*, 1-*OPT* e 2-*OPT* em  $sol$ . As rotas da solução resultante são inseridas no  $pool$  (linha 12) e, quando o laço das linhas 6-12 termina, o resultado do método é retornado na linha 13. Devido ao fato das colunas, no modelo de SP, não conterem informação sobre a ordem de visitação dos clientes e a disposição dos itens relacionados no carregamento do veículo, uma estrutura de dados que armazena essas informações é mantida paralelamente.

---

**Algoritmo 5:** Método *Gera Pool Inicial de Colunas*


---

**Entrada:**  $max\_cols$ .

**Saída:**  $pool$  de colunas para o RLMP.

- 1  $pool \leftarrow \emptyset$ ;
  - 2  $HC_1 \leftarrow NearestCustomer$ ,  $HC_2 \leftarrow NearestNeighbor$ ,  
 $HC_3 \leftarrow BestFitDecreasing$ ,  $HC_4 \leftarrow SmallestLongEdge$ ;
  - 3  $sol_1 \leftarrow$  Heurística(*ILS-1*);
  - 4  $sol_2 \leftarrow$  Heurística(*ILS-2*);
  - 5 Adicione as rotas de  $sol_1$  e  $sol_2$  ao  $pool$ ;
  - 6 **enquanto**  $|pool| < max\_cols$  **faça**
  - 7      $i \leftarrow$  Escolha um valor aleatoriamente no intervalo  $[1, 4]$ ;
  - 8      $sol \leftarrow$  GereSoluçãoInicial( $HC_i$ );
  - 9      $sol \leftarrow$  Perturbação(*Exchange*,  $sol$ );
  - 10      $sol \leftarrow$  Perturbação(1-*OPT*,  $sol$ );
  - 11      $sol \leftarrow$  Perturbação(2-*OPT*,  $sol$ );
  - 12     Adicione as rotas de  $sol$  ao  $pool$ ;
  - 13 **retorna**  $pool$ ;
-

### 5.1.2 O Subproblema de Precificação e a Geração de Novas Colunas

O subproblema de precificação que surge no contexto do 2L-CVRP é um problema de caminho mínimo com algumas características especiais. Primeiramente, sua formulação sugere um grafo com custos nas arestas e prêmios nos vértices, o que é equivalente a uma formulação sem prêmios nos vértices, porém, com possíveis custos negativos nas arestas e, provavelmente, ciclos de custo negativo também. Em segundo lugar, o subproblema de precificação está sujeito a restrições adicionais, tais como as relacionadas às capacidades dos veículos, que limitam a quantidade de recursos que podem ser consumidos ao longo de uma rota. Dadas essas características, pode-se observar que a elementaridade de um caminho e a quantidade de recursos que ele consome são propriedades que devem ser explicitamente garantidas pelo problema. Portanto, o subproblema de precificação abordado se torna o Problema de Caminho Mínimo Elementar com Restrições de Recurso (ESPPRC, do inglês *Elementary Shortest Path Problem with Resource Constraints*), que é NP-difícil se o grafo contiver ciclos de custo negativo (Dror [1994]).

A definição geral do ESPPRC é dada da seguinte maneira: Seja  $G = (V, E)$  um grafo, onde  $V$  é o conjunto de vértices composto pelo conjunto  $N$ , representando os clientes, e por dois vértices  $s$  e  $t$ , representando o depósito. Sejam  $\Phi$  o número de recursos considerados e  $\Psi = \langle \Psi^1, \dots, \Psi^\Phi \rangle$  uma tupla de recursos, tal que  $\Psi^\phi \geq 0$  denota a capacidade do recurso  $\phi$ . Um prêmio  $\pi_i$  é associado a cada vértice  $i \in N$ , um custo  $\tau$  é associado ao depósito e um custo  $c_{ij}$  é associado a cada aresta  $(i, j) \in E$ . Um caminho deve ser traçado de  $s$  a  $t$ , visitando um subconjunto de vértices em  $N$  sem formar ciclos. O objetivo do problema é minimizar o custo total desse caminho, dado pela soma dos custos das arestas menos a soma dos prêmios coletados, satisfazendo as restrições de capacidade para todos os recursos.

Os métodos de resolução do ESPPRC com ciclos de custo negativo normalmente se baseiam em programação dinâmica. Nessa linha, Feillet et al. [2004] propuseram o primeiro algoritmo para resolver o problema, baseando-se no algoritmo de correção de rótulos de Desrochers [1988] para o Problema de Caminho Mínimo com Restrições de Recurso (SPPRC, do inglês *Shortest Path Problem with Resource Constraints*). A ideia geral daquele algoritmo é estender os caminhos vértice a vértice, sem permitir ciclos, por meio de uma busca em largura, onde, através de regras de dominância, os caminhos que levam a soluções sub-ótimas são eliminados, reduzindo drasticamente o tamanho da árvore de enumeração. Outra abordagem, proposta por Boland et al. [2006], aplica o mesmo método explicado anteriormente, contudo, em um primeiro momento ciclos são permitidos. Caso uma solução possua ciclo, os vértices que o compõem são forçados a serem visitados somente uma vez e o algoritmo é executado outra vez com essa condição. O processo é repetido até que um caminho elementar seja encontrado. Por fim, Righini &

Salani [2006] propuseram um algoritmo de busca bidirecional com podas baseado naquele de Feillet et al. [2004]. Nessa abordagem os caminhos são estendidos a partir do vértice de origem e de destino, de modo que uma solução para o problema é obtida quando os caminhos vindos de direções opostas se encontram no mesmo vértice.

No presente trabalho, o subproblema de precificação é resolvido por meio de heurísticas construtivas e da programação dinâmica de Feillet et al. [2004]. As características dessas duas abordagens são explicadas em 5.1.2.1 e 5.1.2.2, enquanto em 5.1.2.3 é apresentado o método que reúne essas abordagens para gerar as colunas a serem inseridas no MP.

### 5.1.2.1 Programação Dinâmica

O algoritmo de programação dinâmica de Feillet et al. [2004] consiste em estender caminhos, a partir de  $s$ , por meio da adição de vértices não visitados ao final desses caminhos. A cada vértice estendido  $i$  são atribuídos rótulos, representando o caminho de  $s$  a  $i$ , que contêm: (a) uma tupla  $\psi = \langle \psi^1, \dots, \psi^\Phi \rangle$  com o consumo dos recursos, no qual cada componente  $\psi^\phi$  representa a quantidade de recurso  $\phi$  consumido ao longo do caminho associado, e (b) o custo total relacionado  $C$ . Rótulos diferentes associados a um mesmo vértice  $i$  indicam caminhos viáveis diferentes que chegam até ele. Dessa forma, um rótulo de um caminho  $p$  associado a um vértice  $i$  tem a forma  $l_p = (\psi(p), C(p), i)$ .

Nesta dissertação, os rótulos se utilizam de três recursos. O primeiro deles ( $\psi^1$ ) é responsável por garantir a elementaridade do caminho, e foi proposto por Beasley & Christofides [1989]. Esse recurso falso é representado por um vetor  $\nu$ , no qual cada posição indica se um vértice  $i \in N$  foi visitado (valor 1) ou não (valor 0). Cada posição de  $\nu$  é inicializada com 0 e seu valor máximo é 1, ou seja,  $\Psi^1 = 1$ . Os outros dois recursos ( $\psi^2$  e  $\psi^3$ ) representam as capacidades do veículo em peso ( $d$ ) e área ( $a$ ). Seus valores são inicializados em 0, considerando que o veículo sai do depósito vazio e coleta os itens no decorrer do caminho, e suas capacidades são  $\Psi^2 = D$  e  $\Psi^3 = A$ . A cada vértice  $i \in N$  visitado, a demanda em peso ( $d_i$ ) e área ( $a_i$ ) do mesmo são somados aos recursos consumidos pelo caminho, respectivamente,  $d$  e  $a$ .

Com base nos dados descritos anteriormente, pode-se definir o rótulo do subproblema de precificação do 2L-CVRP, do caminho  $p$  no vértice  $i$ , como:  $l_p = (\nu(p), d(p), a(p), C(p), i)$ . Quando esse rótulo é estendido para um vértice  $j$ , gerando um outro rótulo viável  $l_{p'} = (\nu(p'), d(p'), a(p'), C(p'), j)$ , o custo, a distância e os recursos desse novo rótulo devem ser calculados e aqueles rótulos em que um ou mais recursos são excedidos devem ser podados. O custo é inicializado com  $-\tau$  e sua atualização é de acordo com a fórmula,

$$C(p') := C(p) - c_{it} + c_{ij} + c_{jt} - \pi_j,$$

que representa o custo reduzido de uma rota, uma vez que  $c_{st} = 0$ . Por outro lado, os recursos são atualizados segundo as fórmulas

$$\nu^k(p') = \begin{cases} \nu^k(p) + 1 & , k = j \\ \nu^k(p) & , k \neq j, \end{cases}$$

$$d(p') := d(p) + d_j \quad \text{e} \quad a(p') := a(p) + a_j.$$

É fácil observar que o número de rótulos que podem ser criados pela programação dinâmica é igual ao número de caminhos possíveis de  $s$  a  $t$ . Portanto, a eficiência desse algoritmo está intimamente ligada à eliminação de rótulos viáveis que não levam a soluções ótimas. Para isso, regras de dominância e de poda foram desenvolvidas para determinar e eliminar esses caminhos.

**Proposição 1.** *Sejam os rótulos  $l_p = (\nu(p), d(p), a(p), C(p), i)$  e  $l_{p'} = (\nu(p'), d(p'), a(p'), C(p'), i)$  associados aos caminhos  $p$  e  $p'$ , respectivamente, que terminam no vértice  $i$ . Se  $\nu^j(p) \leq \nu^j(p')$ ,  $\forall j \in V$ , e  $C(p) \leq C(p')$ , então  $l_p$  domina  $l_{p'}$ .*

*Demonstração.* (Baseado na prova de Lopes [2012]) Sejam  $j$  um vértice qualquer para o qual será considerada a extensão dos rótulos  $l_p$  e  $l_{p'}$ , e  $c_{ij}$  o custo da aresta entre  $i$  e  $j$ . Considere que  $\nu^j(p) = 0$  e  $\nu^j(p') = 0$ , isto é, o vértice  $j$  não foi visitado por  $p$  e nem por  $p'$ . Como resultado, tanto a extensão de  $l_p$  quanto a de  $l_{p'}$  ao vértice  $j$  é possível. Com isso, a extensão dos rótulos  $l_p$  e  $l_{p'}$  ao vértice  $j$  em termos de custo resulta em  $C(p) - c_{it} + c_{ij} + c_{jt} - \pi_j \leq C(p') - c_{it} + c_{ij} + c_{jt} - \pi_j$ . Portanto, o caminho  $p'$  pode ser eliminado, já que sua extensão resulta em um caminho com custo sempre maior ou igual a  $C(p)$ .<sup>1</sup>  $\square$

Observa-se que é possível identificar vértices que não podem ser visitados por nenhuma extensão de um dado rótulo, devido a limitações de recursos. Esses vértices são denominados inalcançáveis. Um vértice  $j$  é considerado inalcançável a partir de um rótulo  $l_p = (\nu(p), d(p), a(p), C(p), i)$  se existe um recurso  $\phi$  que é não decrescente, representado por  $d$  e  $a$ , nesse caso, e dada a extensão do rótulo  $l_p$  para o vértice  $j$ , o consumo do recurso  $\phi$  excede a capacidade máxima disponível. Isso significa que qualquer extensão gerada a partir do rótulo  $l_p$  não conseguirá ser estendida ao vértice  $j$ . Dessa forma, o vértice  $j$  pode ser marcado como visitado no rótulo  $l_p$  ( $\nu^j(p) := 1$ ). Esse mecanismo de poda permite o algoritmo eliminar um grande número de rótulos e diminuir o tempo de execução.

<sup>1</sup>O caso onde  $\nu^j(p) = 0$  e  $\nu^j(p') = 1$  não precisa ser considerado, pois a demonstração para o caso  $\nu^j(p) = 0$  e  $\nu^j(p') = 0$  já é suficiente para mostrar que toda extensão possível do caminho  $p'$  para um vértice que ele ainda não visitou é menos vantajosa que a extensão do caminho  $p$  para o mesmo vértice.

O algoritmo de programação dinâmica desenvolvido tem a execução interrompida quando a primeira solução com custo reduzido negativo é encontrada. Em testes preliminares essa estratégia obteve menores tempos computacionais ao se resolver o subproblema de precificação.

### 5.1.2.2 Heurísticas Construtivas

O processo de resolução exato do subproblema de precificação tende a ser uma tarefa que consome muito tempo computacional, principalmente pelo fato de envolver a resolução de um problema  $\mathcal{NP}$ -difícil. Por isso, métodos heurísticos são normalmente usados para tentar postergar ao máximo a necessidade de utilização do método exato. As heurísticas implementadas executam sobre um grafo auxiliar onde cada aresta  $(i, j)$  tem um custo igual a  $c_{ij} - \pi_i/2 - \pi_j/2$ , tal que  $\pi_s = \tau$  e  $\pi_t = \tau$ . Dessa forma, o custo total da solução obtida é igual ao custo reduzido da rota. As quatro heurísticas desenvolvidas são apresentadas a seguir, na ordem em que são usadas no algoritmo de CG.

**Deterministic Nearest Customer.** Esta heurística realiza a construção das soluções da mesma forma que a heurística construtiva *Nearest Customer*, usada na Subseção 5.1.1. Com a ressalva de que cada vértice escolhido é inserido na posição que resulte na solução com o menor custo real. O algoritmo termina quando uma solução de custo reduzido negativo que respeite as capacidades do veículo é encontrada ou quando não há mais vértices a serem inseridos.

**Deterministic Nearest Neighbor.** Esta heurística realiza a construção das soluções da mesma forma que a heurística construtiva *Nearest Neighbor*, usada na Subseção 5.1.1. O algoritmo termina quando uma solução de custo reduzido negativo que respeite as capacidades do veículo é encontrada ou quando não há mais vértices a serem inseridos.

**Randomized Nearest Neighbor.** Neste caso, a heurística utiliza do mesmo mecanismo da heurística anterior, contudo, a cada iteração, o vértice a ser inserido é escolhido aleatoriamente entre os quatro que resultam no menor aumento do custo. O algoritmo termina quando uma solução de custo reduzido negativo que respeite as capacidades do veículo é encontrada ou quando não há mais vértices a serem inseridos.

**Kruskal-based Clustering.** Assim como no algoritmo de Kruskal para Árvores Geradoras Mínimas, as arestas são ordenadas em ordem não decrescente e, iterativamente, a partir do início dessa ordenação, as arestas são adicionadas à solução sem formar ciclos. Nesta heurística, especificamente, quando duas componentes conexas se unem por meio de uma aresta inserida, é verificado se o veículo tem capacidade

suficiente para atender os vértices pertencentes às componentes conectadas. Caso a capacidade seja extrapolada, a união não é feita e a próxima aresta é selecionada, caso contrário, a união é realizada. O algoritmo termina se alguma componente conexa tiver custo reduzido negativo, ou quando todas as arestas a serem inseridas se esgotarem. Quando há uma componente conexa com custo reduzido negativo, quer dizer que há uma solução, logo, os vértices associados devem ser roteados de acordo com a heurística *Nearest Neighbor*.

### 5.1.2.3 O Método Completo de Precificação

Devido à complexidade da programação dinâmica, sua execução se limita às vezes que as heurísticas usadas não retornam nenhuma coluna a ser inserida no RLMP. Essa estratégia é detalhada no Algoritmo 6, onde *newCols* é um conjunto que armazena as novas colunas criadas, e  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$  e  $M_5$  se referem respectivamente aos métodos de precificação *Deterministic Nearest Customer*, *Deterministic Nearest Neighbor*, *Randomized Nearest Neighbor*, *Kruskal Based Clustering* e Programação Dinâmica. Na linha 1 o conjunto *newCols* é inicializado como um conjunto vazio e na linha 2 a variável auxiliar é inicializada. O laço das linhas 3-6 realiza a iteração sobre os métodos utilizados. Novas colunas são atribuídas ao conjunto *newCols* na linha 4, e a variável auxiliar é incrementada em uma unidade na linha 5. O laço termina quando pelo menos uma coluna é encontrada ou quando todos os métodos são executados e nenhuma coluna é gerada. O conjunto com as colunas geradas é retornado na linha 7.

---

#### Algoritmo 6: Método *Gera Novas Colunas*

---

**Entrada:** -

**Saída:** Novas colunas a serem inseridas no RLMP.

```

1 newCols  $\leftarrow \emptyset$ ;
2 i  $\leftarrow 1$ ;
3 repita
4   | newCols  $\leftarrow newCols \cup ColunasGeradasPor(M_i)$ ;
5   | i  $\leftarrow i + 1$ ;
6 até  $|newCols| > 0$  ou i > 5;
7 retorna newCols;

```

---

### 5.1.3 Heurística de Geração de Colunas com Verificação da Viabilidade dos Carregamentos na Precificação (CGV)

O primeiro método desenvolvido neste trabalho para resolver o 2L-CVRP é denominado Heurística de Geração de Colunas com Verificação da Viabilidade dos Carregamentos na Precificação (CGV, do inglês *Column Generation with Loading Verification in Pricing*). A sua estrutura segue aquela clássica utilizada pelo algoritmo de CG, apresentada no

Algoritmo 2. Durante a execução do CGV as novas colunas são geradas por meio do método apresentado em 5.1.2.3, entretanto, além de certificar que a capacidade do veículo é respeitada, o método deve garantir também que o carregamento seja viável. Essa operação é realizada pelo método apresentado no Algoritmo 1 sempre que ocorre a tentativa de inserção de um vértice num caminho, na programação dinâmica, ou solução, nas heurísticas construtivas. Vale observar que, devido ao fato da viabilidade ser verificada por um método heurístico, o algoritmo de programação dinâmica é descaracterizado como método exato. Finalmente, quando o processo de CG termina, o RLMP resultante tem suas restrições 5.2 substituídas pelas restrições originais 3.11, dando origem a um problema onde o conjunto de colunas é o mesmo do RLMP e as variáveis são inteiras. Esse novo problema é então resolvido por um algoritmo BB de um resolvidor comercial e a solução final é obtida.

O pseudocódigo do CGV é apresentado no Algoritmo 7. Considera-se *cols* como um conjunto de colunas e *sol* como uma solução inteira para o problema. Na linha 1 ocorre a inicialização do *pool* inicial de colunas. No laço das linhas 2-6 ocorre a geração das novas colunas e a resolução do RLMP, até que nenhuma coluna seja gerada ou até que o critério de parada  $CP_1$  seja atendido. As novas colunas são adicionadas ao modelo na linha 3 e esse é resolvido à otimalidade na linha 4. Em seguida, o conjunto de colunas (*cols*) é esvaziado (linha 5) e possíveis novas colunas são geradas (linha 6), de tal forma que a viabilidade dos carregamentos das rotas/colunas sejam atestadas pelo Algoritmo 1. Na linha 7, o RLMP tem suas variáveis convertidas para inteiro e é resolvido por meio de um algoritmo BB, a fim de obter uma solução inteira. Ao final do algoritmo, o resultado é retornado na linha 8.

---

**Algoritmo 7:** Heurística CGV
 

---

**Entrada:** RLMP,  $CP_1$ .

**Saída:** Solução viável para o 2L-CVRP.

```

1 cols ← Inicialize o pool de colunas;
2 enquanto  $|cols| > 0$  e critério de parada  $CP_1$  não for atendido faça
3   Adicione cols ao RLMP;
4   Resolva(RLMP);
5   cols ←  $\emptyset$ ;
6   cols ← GeraNovasColunas(Com verificação da viabilidade do
   carregamento);
7 sol ← Resolva(RLMP, inteiro, BB);
8 retorna sol;

```

---

### 5.1.4 Heurística de Geração de Colunas sem Verificação da Viabilidade dos Carregamentos na Precificação (CGN)

A Heurística de Geração de Colunas sem Verificação da Viabilidade dos Carregamentos na Precificação (CGN, do inglês *Column Generation without Loading Verification in Pricing*) é o segundo método desenvolvido nesta dissertação para a resolução do 2L-CVRP. Assim como o CGV, sua estrutura é baseada naquela clássica de CG (Algoritmo 2). No caso do CGN, o método para gerar colunas apresentado em 5.1.2.3 é utilizado sem a necessidade de verificar a viabilidade dos carregamentos dos veículos. Ao terminar a geração das colunas, o RLMP resultante tem suas restrições 5.2 substituídas pelas restrições originais 3.11 e é então resolvido por um algoritmo BB. Nesse momento, os carregamentos associados às rotas da solução obtida têm a viabilidade verificada. Primeiramente, a informação sobre a viabilidade dos carregamentos é procurada na árvore de viabilidade, caso não seja encontrada, o Algoritmo 1 e o BB aproximado (Seção 4.3) são executados para tentar obter uma resposta. Caso todos os carregamentos da solução sejam viáveis, o algoritmo termina. Caso contrário, as rotas associadas aos carregamentos inviáveis são removidas do conjunto de colunas do RLMP e proibidas de serem geradas novamente, e o processo de CG é reiniciado com as colunas que sobrarem. É importante notar que o valor final da relaxação linear obtido por essa heurística não representa, necessariamente, um limite inferior para a solução ótima do problema, pois o BB aproximado pode falhar ao verificar a viabilidade de uma rota e, por consequência, eliminar rotas viáveis.

O Algoritmo 8 apresenta o pseudocódigo da heurística CGN. Sejam  $cols$  um conjunto de colunas,  $sol$  uma solução inteira do problema (não necessariamente viável) e  $viab$  uma variável que armazena informação sobre a viabilidade de uma rota. Na linha 1,  $cols$  é inicializado com o  $pool$  inicial de colunas. No laço das linhas 2-20 ocorre o processo de CG e verificação da viabilidade das rotas, até que o critério de parada  $CP_1$  seja atendido. No laço das linhas 3-7 é executado o processo de geração das novas colunas e resolução do RLMP, até que não sejam mais geradas colunas. Na linha 4, as novas colunas são adicionadas ao  $cols$ ; na linha 5, o RLMP é resolvido à otimalidade; na linha 6, o conjunto  $cols$  é esvaziado; e na linha 7, novas colunas são geradas sem que haja verificação da viabilidade dos carregamentos. Na linha 8 o RLMP tem as variáveis transformadas em valores inteiros e é resolvido por um algoritmo BB de um resolvedor comercial. O laço das linhas 9-14 é responsável por verificar a viabilidade de todas as rotas  $r$  de  $sol$ . Caso a informação sobre viabilidade de  $r$  na árvore seja indeterminada (linha 10), a viabilidade da mesma é testada pelo Algoritmo 1 e o resultado é armazenado na variável  $viab$  (linha 11); se  $viab$  for inviável, a viabilidade é verificada novamente pelo algoritmo BB aproximado (Seção 4.3), na linha 13; e finalmente, na linha 14, a rota  $r$  e a informação sobre seu carregamento são adicionados na árvore de viabilidade. Na linha 15 é verificado se todas as rotas de  $sol$  são viáveis e, em caso afirmativo, a solução é retornada pelo algoritmo,



e em caso negativo, o laço das linhas 18-20 remove todas as colunas inviáveis de *sol* do **RLMP** (linha 19) e as proíbe de serem inseridas novamente (linha 20). Essa proibição se dá por meio da utilização da árvore de viabilidade, a qual é preenchida com as rotas inviáveis e posteriormente é consultada no momento da geração de novas colunas (linha 7) para evitar a reinserção dessas rotas como colunas. Caso o laço das linhas 2-20 seja interrompido pelo critério de parada, significa que nenhuma solução viável foi encontrada e, portanto, retorna-se uma solução vazia na linha 21.

---

**Algoritmo 8:** Heurística **CGN**


---

**Entrada:**  $CP_1$ .  
**Saída:** Solução viável para o **2L-CVRP**, quando encontrada.

```

1   $cols \leftarrow$  Inicialize o pool de colunas;
2  enquanto critério de parada  $CP_1$  não for atendido faça
3      enquanto  $|cols| > 0$  faça
4          Adicione  $cols$  ao RLMP;
5          Resolva(RLMP);
6           $cols \leftarrow \emptyset$ ;
7           $cols \leftarrow$  GeraNovasColunas(Sem verificação da viabilidade do
           carregamento);
8   $sol \leftarrow$  Resolva(RLMP, inteiro, BB);
9  para toda rota  $r$  de  $sol$  faça
10     se RecupereInformaçãoNaÁrvore( $r$ ) = indeterminado então
11          $viab \leftarrow$  VerifiqueCarregamento(Coleção de heurísticas para o
           carregamento,  $r$ );
12         se  $viab =$  inviável então
13              $viab \leftarrow$  VerifiqueCarregamento(BB aproximado,  $r$ );
14         InsiraNaÁrvore( $r$ ,  $viab$ );
15 se todas as rotas de  $sol$  são viáveis então
16     retorna  $sol$ ;
17 senão
18     para cada rota inviável  $r$  de  $sol$  faça
19         RemovaColuna(RLMP,  $r$ );
20         ProíbaReinserção( $r$ );
21 retorna  $\emptyset$ ;
```

---

## 5.2 *Branch-and-Cut*

O algoritmo **BC** é um método que combina o algoritmo de planos de corte com o processo de enumeração do **BB**. Esse método é utilizado para resolver Problemas de Programação Linear Inteira com um número exponencial de restrições. Inicialmente, o **BC** resolve, via Método Simplex, a relaxação linear do problema original sem as grandes famílias de

restrições 3.4 e 3.5. A medida que restrições violadas dessas famílias são identificadas, cortes são inseridos no modelo relaxado. Quando não há mais cortes a serem inseridos e a solução não é inteira, o método BB se aplica e o processo de inserção de cortes se repete nos outros nós da árvore de enumeração. O algoritmo termina quando o valor da melhor solução inteira encontrada for igual ao valor da relaxação linear.

Neste trabalho, um algoritmo BC baseado naquele proposto por Iori et al. [2007] é utilizado para resolver o problema definido pelo modelo adaptado de Fluxo de Veículo (Seção 3.1) para o caso onde (a) o número de rotas utilizadas pode ser menor que o número de veículos e (b) rotas com um consumidor são permitidas. A adaptação para (a) consiste na substituição da restrição 3.3 por

$$\sum_{(i,j) \in \delta(0)} z_{ij} \leq 2K, \quad (5.5)$$

enquanto para (b) se trata apenas de permitir que  $z_{ij}$  assumam valor 2 quando  $(i, j) \in \delta(0)$ . A necessidade de utilizar o algoritmo BC para resolver o modelo utilizado se dá pelo fato de mesmo conter as grandes famílias de restrições 3.4 (CCCs) e 3.5 (IPCs).

Devido ao fato de  $r(S)$  envolver a resolução de um problema NP-difícil, o problema de separação associado às restrições 3.4 também o é (Naddef & Rinaldi [2001]). Portanto, são necessárias boas heurísticas para tentar encontrar restrições violadas por uma solução  $z^*$ , uma vez que é improvável obter um algoritmo exato para resolver esse problema em tempo polinomial. Os três procedimentos a seguir são adaptações feitas por Iori et al. [2007] a partir de heurísticas simples usadas para o mesmo fim no CVRP.

**Procedimento 1.** Seja  $G^* = (V, E^*)$  o grafo suporte de  $z^*$  definido por  $E^* := \{(i, j) \in E : z_{ij}^* > 0\}$ , onde a capacidade associada a cada aresta  $(i, j)$  é representada por  $z_{ij}^*$ . Para cada cliente  $i$ , deve-se encontrar o corte mínimo  $\delta(S^*)$  que separa o depósito de  $i$  e tal que  $i \in S^*$ . Em seguida, verifica-se a restrição 3.4 definida por  $S := S^*$  para o caso de uma possível violação com respeito à relaxação  $r'(S)$ .

**Procedimento 2.** Seja  $n + 1$  um vértice falso e  $G' = (V \cup \{n + 1\}, E')$  um grafo, onde  $E'$  contém todas as arestas em  $E^*$  com as mesmas capacidades do Procedimento 1, mais uma nova aresta conectando cada cliente  $i$  e  $n + 1$  com capacidade  $2d_i/D$ . O objetivo é encontrar o corte mínimo separando o depósito de  $n + 1$ , tal que  $n + 1 \in S'$ , e verificar a restrição 3.4 definida por  $S := S' \setminus \{n + 1\}$ .

**Procedimento 3.** Realiza o mesmo que o Procedimento 2, contudo, com capacidades  $2a_i/A$  nas arestas falsas.

O procedimento de separação das restrições 3.5 está relacionado à verificação da viabilidade dos carregamentos associados às rotas. O Algoritmo 9 exibe o pseudocódigo

do método utilizado para verificar os carregamentos dessas rotas. Seja  $viab$  uma variável que armazena informação sobre a viabilidade de uma rota  $r$  analisada. Na linha 1,  $viab$  é inicializada como viável. Na linha 2 é verificada a informação sobre o carregamento de  $r$  na árvore de viabilidade. Caso a informação seja indeterminada, a viabilidade é testada pelo Algoritmo 1, pelo **BB** aproximado (Seção 4.3) e pelo **BB** (Seção 4.2), nas linhas 3, 4 e 5, respectivamente. Se nenhum desses algoritmos atestar a viabilidade do carregamento de  $r$ , o mesmo é considerado inviável e a variável  $viab$  armazena essa informação (linha 6). Caso a informação sobre  $r$  na árvore de viabilidade, seja que o carregamento é inviável (linha 7), então a variável  $viab$  é atualizada (linha 8). Finalmente, o resultado é retornado na linha 9. Por se tratar de um método muito custoso, com relação ao tempo computacional, esse só é executado quando os Procedimentos 1-3 não retornam nenhuma restrição violada e a solução  $z^*$  é inteira. O Procedimento 4 define esse mecanismo de separação.

---

**Algoritmo 9:** Método *Verifica Carregamento BC*


---

**Entrada:**  $r$ .  
**Saída:** Situação do carregamento (viável ou inviável) referente a rota  $r$ .

- 1  $viab \leftarrow$  viável;
- 2 **se**  $RecupereInformaçãoNaÁrvore(r) = indeterminado$  **então**
- 3     **se**  $VerifiqueCarregamento(Coleção\ de\ heurísticas\ para\ o\ carregamento, r) =$   
        $inviável$  **então**
- 4         **se**  $VerifiqueCarregamento(BB\ aproximado, r) = inviável$  **então**
- 5             **se**  $VerifiqueCarregamento(BB, r) = inviável$  **então**
- 6                  $viab \leftarrow$  inviável;
- 7 **se**  $RecupereInformaçãoNaÁrvore(r) = inviável$  **então**
- 8      $viab \leftarrow$  inviável;
- 9 **retorna**  $viab$ ;

---

**Procedimento 4.** Para cada rota  $(S, \sigma)$  da solução  $z^*$ , a viabilidade de seu carregamento associado é verificada. A verificação é realizada pelo método apresentado no Algoritmo 9. Caso o carregamento de  $(S, \sigma)$  seja inviável, a restrição 3.5 associada é inserida no modelo. Por outro lado, se todas as rotas da solução forem viáveis, a solução inteira corrente é atualizada.



# Capítulo 6

## Experimentos Computacionais

Neste capítulo são apresentados os detalhes dos experimentos computacionais realizados para avaliar a qualidade dos algoritmos desenvolvidos. Na Seção 6.1 são descritos o ambiente computacional, as instâncias de teste e as características do problema que foram consideradas. A Seção 6.2 mostra como foi feita a geração do *pool* inicial de colunas para o CGV e o CGN, além de exibir como foram configurados os parâmetros dos algoritmos. Por fim, as Seções 6.3 e 6.4 apresentam os resultados obtidos pelos métodos testados, onde a primeira relata os ganhos consequentes da utilização da árvore de viabilidade, e a última faz a comparação com os resultados da literatura.

### 6.1 Ambiente Computacional, Instâncias de Teste e Características Consideradas

Os experimentos foram realizados em duas máquinas distintas, onde uma delas foi usada pelos algoritmos CGV e CGN, e a outra, pelo método BC. A primeira delas conta com um processador Intel® Core™ i7 com 3.33 GHz e 24 GB de memória RAM, executando o sistema operacional Ubuntu 12.04. Já a segunda, utiliza um processador Intel® Xeon® E5645 com 2.4 GHz e 32 GB de memória RAM, rodando também sob o sistema operacional Ubuntu 12.04. Os algoritmos foram implementados em C++ e compilados com o GCC 4.5.2. A plataforma ILOG CPLEX 12.5 foi utilizada como resolvidor de Programação Linear para todos os algoritmos desenvolvidos e também como gerenciador da árvore de BB para o método BC. Foi permitida a utilização de 1 núcleo do processador para que essa plataforma executasse suas funções.

Para avaliar o desempenho dos algoritmos propostos, foram usadas as 180 instâncias do 2L-CVRP, propostas por Iori et al. [2007]. Essas instâncias são uma extensão de outras 36, clássicas do VRPTW (Solomon [1987]), de tal forma que as coordenadas originais e as demandas  $d_i$  dos clientes são preservadas, e a informação adicional incluída é expressa

pela demanda dos clientes por itens bidimensionais retangulares com dimensões variadas. A largura ( $W$ ) e o comprimento ( $L$ ) da superfície do compartimento de carregamento dos veículos, definidos para todas as instâncias, são iguais a 20 e 40, respectivamente. Os custos das arestas se referem às distâncias Euclidianas entre os pares de clientes no grafo. Para gerar as instâncias descritas, cinco classes de demandas por itens foram introduzidas por Iori et al. [2007].

**Classe 1.** Para cada cliente é associado um único item de largura e comprimento iguais a um. Os problemas desta classe são, na verdade, instâncias puras do CVRP, ou seja, toda sequência de clientes é viável em termos das restrições de carregamento. Essas instâncias são usadas para testar a efetividade dos algoritmos com relação ao roteamento.

**Classes 2-5.** Para cada cliente  $i$ , um conjunto de  $m_i$  itens é gerado, tal que  $m_i$  é um valor uniformemente distribuído em um dado intervalo. Por sua vez, cada item é classificado em uma de três possíveis formas, nomeadas Verticais, Homogêneas e Horizontais, com a mesma probabilidade. As dimensões (largura e comprimento) de um desses itens são distribuídas uniformemente em intervalos determinados pela sua forma. Os intervalos de valores para  $m_i$  e para as dimensões dos itens são apresentados na Tabela 6.1.

**Tabela 6.1:** Características da geração dos itens.

Cl	Verticais			Homogêneas		Horizontais	
	$m_i$	$l_i$	$w_i$	$l_i$	$w_i$	$l_i$	$w_i$
2	[1,2]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]
3	[1,3]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]
4	[1,4]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]
5	[1,5]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]	[0,4L, 0,9L]	[0,1W, 0,2W]

Para cada um dos 36 problemas do CVRP, cinco instâncias foram criadas de acordo com as cinco classes apresentadas (uma instância por classe), resultando nas 180 instâncias para o 2L-CVRP. A Tabela 6.2 apresenta as características dessas instâncias para as Classes 1-5. As duas primeiras colunas apresentam o número do problema e o tamanho do conjunto  $V$  de cada instância. As colunas 3 e 4 apresentam o número total de itens ( $M = \sum_{i \in V \setminus \{0\}} m_i$ ) e o tamanho da frota de veículos disponíveis para cada instância da Classe 1. Os mesmos dados são apresentados para as Classes 2-5, respectivamente, nas colunas 5-6, 7-8, 9-10 e 11-12. Vale observar que o número de veículos disponíveis para as instâncias das Classes 2-5 foi determinado por meio de heurísticas para o 2BPP.

Para a execução dos testes foram considerados dois conjuntos de características possíveis. O primeiro representa as instâncias com as características originais, usadas para realizar os testes do primeiro algoritmo desenvolvido para o 2L-CVRP, no caso,

**Tabela 6.2:** Características das instâncias.

I	V	Classe 1		Classe 2		Classe 3		Classe 4		Classe 5	
		M	K	M	K	M	K	M	K	M	K
1	16	15	3	24	3	31	3	37	4	45	4
2	16	15	5	25	5	31	5	40	5	48	5
3	21	20	4	29	5	46	5	44	5	49	5
4	21	20	6	32	6	43	6	50	6	62	6
5	22	21	4	31	4	37	4	41	4	57	5
6	22	21	6	33	6	40	6	57	6	56	6
7	23	22	3	32	5	41	5	51	5	55	6
8	23	22	5	29	5	42	5	48	5	52	6
9	26	25	8	40	8	61	8	63	8	91	8
10	30	29	3	43	6	49	6	72	7	86	7
11	30	29	4	43	6	62	7	74	7	91	7
12	31	30	9	50	9	56	9	82	9	101	9
13	33	32	3	44	7	56	7	78	7	102	8
14	33	32	4	47	7	57	7	65	7	87	8
15	33	32	5	48	6	59	6	84	8	114	8
16	36	35	11	56	11	74	11	93	11	114	11
17	41	40	14	60	14	73	14	96	14	127	14
18	45	44	4	66	9	87	10	112	10	122	10
19	51	50	5	82	11	103	11	134	12	157	12
20	72	71	4	104	14	151	15	178	16	226	16
21	76	75	7	114	14	164	17	168	17	202	17
22	76	75	8	112	15	154	16	198	17	236	17
23	76	75	10	112	14	155	16	179	16	225	16
24	76	75	14	124	17	152	17	195	17	215	17
25	101	100	8	157	21	212	21	254	22	311	22
26	101	100	10	147	19	198	20	247	20	310	20
27	101	100	14	152	19	211	22	245	22	320	22
28	121	120	7	183	23	242	25	299	25	384	25
29	135	134	7	197	24	262	26	342	28	422	28
30	151	150	12	225	29	298	30	366	30	433	30
31	200	199	16	307	38	402	40	513	42	602	42
32	200	199	17	299	38	404	39	497	39	589	39
33	200	199	17	301	37	407	41	499	41	577	41
34	241	240	22	370	46	490	49	604	50	720	50
35	253	252	27	367	45	507	50	634	50	762	50
36	256	255	14	387	47	511	51	606	51	786	51

o BC de Iori et al. [2007]. Essas características são as seguintes: (a) as distâncias Euclidianas entre os clientes são truncadas, (b) o número mínimo de clientes por rota é igual a dois, e (c) o número de veículos utilizados deve ser igual ao tamanho da frota disponível. Por outro lado, o segundo conjunto de características é aquele que foi adotado pela maioria dos trabalhos que sucederam o trabalho pioneiro para o 2L-CVRP. Essas características são dadas por: (a) as distâncias entre os clientes assumem os valores reais, (b) permite-se rotas com apenas um cliente, e (c) podem ser utilizados menos veículos que a quantidade disponível. De modo a simplificar a referência a essas características, o primeiro conjunto foi denominado *Características Ti*, e o segundo, *Características Tr*.

## 6.2 Configuração dos Algoritmos

O *pool* de colunas utilizado pelos algoritmos **CGV** e **CGN** foi gerado a priori pelo Algoritmo 5. Seu tamanho foi fixado em 1000 (*max\_cols*) para todas as instâncias, e suas colunas foram utilizadas pelos algoritmos que consideram as duas *Características*. O valor de *max\_cols* foi definido de forma que a variabilidade do *pool* inicial fosse garantida e, ao mesmo tempo, não sobrecarregasse o **RLMP**. Contudo, essa geração se baseou apenas nas *Características*  $T_i$ , uma vez que suas soluções consistem em soluções viáveis para as *Características*  $Tr$ , sendo necessário apenas converter os valores das distâncias para reais. No método utilizado, os parâmetros considerados pelos algoritmos ILS-1 e ILS-2 foram definidos de tal modo que a execução dos mesmos requeresse baixo tempo computacional. Dessa forma, após algumas combinações de parâmetros testadas, as escolhidas foram as seguintes: (a) no ILS-1, os valores 20, 10 e 50 iterações foram atribuídos aos parâmetros  $CP_1$ ,  $CP_2$  e  $CP_3$ , respectivamente, enquanto (b) no ILS-2, os parâmetros  $CP_1$  e  $CP_2$  receberam os valores 20 e 50 iterações, respectivamente .

O critério de parada utilizado pelos algoritmos propostos foi o tempo de execução, que é o mesmo utilizado pelos métodos da literatura do **2L-CVRP**. O tempo máximo de execução da heurística **CGV** foi baseado nos limites impostos às heurísticas da literatura, isto é, cada instância foi executada por, no máximo, 3600 segundos (garantido pelo critério de parada  $CP_1$ , Algoritmo 7). Por outro lado, o **CGN**, que retorna uma solução viável somente ao fim de sua execução, teve seu limite de tempo estendido para 14400 segundos (garantido pelo critério de parada  $CP_1$ , Algoritmo 8). Contudo, a partir da instância 21, os algoritmos de precificação representam mais de 99% do tempo total de execução do **CGN**. Por isso, o tempo máximo de execução dessas instâncias foi reduzido para 3600 segundos. Por fim, o algoritmo **BC** usou o mesmo limite de tempo utilizado pelo **BC** de Iori et al. [2007], ou seja, 86400 segundos para o método completo e 7200 segundos para o **BB** exato. Com relação ao **BB** Aproximado, os algoritmos **CGN** e **BC** o usam com as configurações  $\mu = 10$  e 600 segundos de tempo máximo de execução. Essas configurações foram definidas para que o **BB** Aproximado obtivesse, em um tempo razoável, resultados com baixa taxa de falso-negativos.

## 6.3 Ganho de Desempenho com a Árvore de Viabilidade

A árvore de viabilidade é usada pelos três algoritmos desenvolvidos neste trabalho. Em especial, o **CGV** é o método que mais utiliza a estrutura, pois a cada cliente inserido em uma rota, no momento da precificação, resulta em uma verificação da viabilidade. Por isso, a análise da eficiência da árvore foi conduzida sobre os resultados desse algoritmo.



Para verificar o desempenho da árvore de viabilidade, dois fatores foram analisados: (a) quantas vezes a utilização da árvore foi útil, ou seja, das vezes que foi solicitada, em quantas foi capaz de definir se um carregamento era viável ou não, e (b) qual parcela do tempo de verificação dos carregamentos foi utilizada por ela. Dados esses fatores, pôde ser analisado que, no **CGV**, a árvore foi útil em 86,87% das vezes no caso sequencial e em 94,05% das vezes no caso irrestrito. Vale lembrar que consultas úteis à árvore não se resumem somente ao caso de carregamentos previamente avaliados pelos algoritmos, existem também os casos onde a viabilidade dos carregamentos podem ser inferidas devido às Propriedades apresentadas na Subseção 4.4.1. Além disso, as tentativas de recuperação de informação na árvore representaram, respectivamente, 4,91% e 14,35% do tempo total de verificação dos carregamentos nos casos sequencial e irrestrito. Esses dados indicam que a utilização da árvore pode proporcionar reduções de cerca de 90% no tempo computacional do algoritmo.

## 6.4 Desempenho dos Algoritmos Desenvolvidos

Esta Seção apresenta as comparações dos resultados dos algoritmos propostos com aqueles da literatura. Na Subseção 6.4.1, são apresentadas as comparações considerando-se as *Características Ti*, enquanto na Subseção 6.4.2, considera-se as *Características Tr*. A fim de economizar espaço em algumas tabelas comparativas, os resultados das instâncias da Classe 1 para os algoritmos **CGV** e **CGN** foram agrupados na mesma coluna, uma vez que é fácil observar que para essa Classe os dois algoritmos tem o comportamento idêntico. Todas as tabelas criadas exibem os resultados desde a instância 1 até àquela que o algoritmo **CGN** consegue obter uma solução viável nas Classes 2-5. Os resultados detalhados dos três algoritmos implementados podem ser vistos no Apêndice A.

### 6.4.1 Resultados para as *Características Ti*

As tabelas 6.3, 6.4 e 6.5 apresentam os resultados obtidos pelos algoritmos **CGV**, **CGN**, e por aqueles propostos na literatura que respeitam as *Características Ti*. Os algoritmos da literatura considerados nas comparações são: o **BC** de Iori et al. [2007] ( $BC_{ior}$ ), o **BC** de Azevedo et al. [2009] ( $BC_{aze}$ ), a Busca Tabu de Gendreau et al. [2008] ( $TS_{gen}$ ), e a Otimização de Colônia de Formigas de Fuellerer et al. [2009] ( $ACO_{fue}$ ). Nessas três tabelas, os melhores resultados de cada instância estão destacados em negrito, os ótimos dos algoritmos **BC** estão sublinhados, e os valores reportados para o  $ACO_{fue}$  se referem às médias de 10 execuções do mesmo.

Na Tabela 6.3 apresentam-se os valores das soluções obtidas por quatro algoritmos da literatura e pelo **CGV** e **CGN**, para as instâncias da Classe 1. A coluna 1 indica qual a instância analisada. As colunas 2-5 apresentam os resultados obtidos pelos algoritmos da

literatura, conforme a nomenclatura explicada anteriormente. Por fim, a última coluna agrupa os resultados do **CGV** e **CGN**. Vale ressaltar que o resultado obtido pelo  $BC_{aze}$  para a instância 13 parece estar errado, uma vez que o valor foi menor que o valor ótimo obtido pelo  $BC_{ior}$ . Pode-se observar que das 17 instâncias analisadas, os algoritmos propostos conseguiram alcançar o melhor resultado em 13 delas, sendo que nas outras, a distância percentual média dos resultados obtidos daqueles melhores foi de 1,77%.

**Tabela 6.3:** Comparação das instâncias da Classe 1 com as *Características Ti*.

I	$BC_{ior}$	$BC_{aze}$	$TS_{gen}$	$ACO_{fue}$	$CGV/CGN$
1	<b>273</b>	<b>273</b>	<b>273</b>	<b>273,0</b>	<b>273</b>
2	<b>329</b>	<b>329</b>	<b>329</b>	<b>329,0</b>	<b>329</b>
3	<b>351</b>	<b>351</b>	<b>351</b>	<b>351,0</b>	362
4	<b>423</b>	<b>423</b>	<b>423</b>	<b>423,0</b>	<b>423</b>
5	<b>367</b>	<b>367</b>	<b>367</b>	<b>367,0</b>	<b>367</b>
6	<b>488</b>	<b>488</b>	<b>488</b>	<b>488,0</b>	<b>488</b>
7	<b>558</b>	<b>558</b>	<b>558</b>	<b>558,0</b>	<b>558</b>
8	<b>657</b>	<b>657</b>	<b>657</b>	664,6	<b>657</b>
9	<b>609</b>	<b>609</b>	<b>609</b>	<b>609,0</b>	<b>609</b>
10	<b>524</b>	<b>524</b>	544	524,3	527
11	<b>500</b>	<b>500</b>	<b>500</b>	<b>500,0</b>	<b>500</b>
12	599	<b>596</b>	598	598,6	<b>596</b>
13	<b>1991</b>	1909*	<b>1991</b>	<b>1991,0</b>	<b>1991</b>
14	827	<b>823</b>	<b>823</b>	823,9	845
15	<b>907</b>	<b>907</b>	<b>907</b>	<b>907,0</b>	<b>907</b>
16	<b>682</b>	<b>682</b>	<b>682</b>	<b>682,0</b>	688
17	859	-	<b>842</b>	844,1	<b>842</b>

A Tabela 6.4, por sua vez, apresenta os resultados obtidos pelos mesmos algoritmos citados na tabela anterior para as instâncias das Classes 2-5 no caso sequencial. A primeira e a segunda colunas representam o número da instância e a classe a que pertence. As colunas 3-6 apresentam os resultados obtidos pelos algoritmos da literatura, enquanto as colunas 7 e 8 apresentam os resultados do **CGV** e do **CGN**. Das 68 instâncias consideradas o **CGV** obteve 16 melhores resultados, enquanto o **CGN** obteve 30. Entre as instâncias que esses algoritmos não conseguiram o melhor resultado, a distância percentual média entre os valores obtidos e os melhores são de 2,75% para o primeiro e 1,97% para o segundo.

**Tabela 6.4:** Comparação das instâncias das Classes 2-5 com as *Características Ti* no caso sequencial.

I	Cl	$BC_{ior}$	$BC_{aze}$	$TS_{gen}$	$ACO_{fue}$	$CGV$	$CGN$
1	2	<b>285</b>	<b>285</b>	<b>285</b>	288,6	313	<b>285</b>
	3	<b>280</b>	<b>280</b>	<b>280</b>	300,9	310	315
	4	<b>288</b>	<b>288</b>	290	290,6	290	290
	5	<b>279</b>	<b>279</b>	<b>279</b>	<b>279,0</b>	<b>279</b>	<b>279</b>
2	2	<b>342</b>	<b>342</b>	<b>342</b>	<b>342,0</b>	347	347
	3	<b>347</b>	<b>347</b>	350	350,6	350	350

I	CI	$BC_{ior}$	$BC_{aze}$	$TS_{gen}$	$ACO_{fue}$	$CGV$	$CGN$
	4	<b>336</b>	<b>336</b>	<b>336</b>	336,3	<b>336</b>	<b>336</b>
	5	<b>329</b>	<b>329</b>	<b>329</b>	<b>329,0</b>	<b>329</b>	<b>329</b>
3	2	<b>389</b>	396	407	396,0	407	405
	3	<b>387</b>	<b>387</b>	<b>387</b>	392,4	409	393
	4	<b>374</b>	<b>374</b>	<b>374</b>	374,8	377	<b>374</b>
	5	<b>369</b>	<b>369</b>	<b>369</b>	<b>369,0</b>	<b>369</b>	<b>369</b>
4	2	<b>434</b>	<b>434</b>	<b>434</b>	<b>434,0</b>	445	<b>434</b>
	3	<b>432</b>	<b>432</b>	438	437,0	438	434
	4	<b>438</b>	<b>438</b>	451	<b>438,0</b>	<b>438</b>	<b>438</b>
	5	<b>423</b>	<b>423</b>	<b>423</b>	<b>423,0</b>	<b>423</b>	<b>423</b>
5	2	<b>380</b>	<b>380</b>	396	<b>380,0</b>	385	382
	3	<b>373</b>	<b>373</b>	377	375,0	389	<b>373</b>
	4	<b>377</b>	<b>377</b>	406	378,8	400	<b>377</b>
	5	<b>389</b>	<b>389</b>	<b>389</b>	<b>389,0</b>	<b>389</b>	<b>389</b>
6	2	<b>491</b>	<b>491</b>	498	493,8	<b>491</b>	<b>491</b>
	3	<b>496</b>	<b>496</b>	<b>496</b>	496,8	<b>496</b>	<b>496</b>
	4	<b>489</b>	<b>489</b>	503	492,0	503	495
	5	<b>488</b>	<b>488</b>	<b>488</b>	<b>488,0</b>	494	492
7	2	<b>724</b>	<b>724</b>	752	<b>724,0</b>	757	748
	3	<b>698</b>	<b>698</b>	704	702,6	<b>698</b>	<b>698</b>
	4	<b>714</b>	<b>714</b>	742	722,0	736	723
	5	<b>742</b>	<b>742</b>	743	743,6	746	743
8	2	<b>720</b>	<b>720</b>	<b>720</b>	<b>720,0</b>	<b>720</b>	<b>720</b>
	3	<b>730</b>	<b>730</b>	752	730,3	763	738
	4	<b>701</b>	<b>701</b>	722	721,7	726	717
	5	<b>721</b>	<b>721</b>	736	729,1	736	<b>721</b>
9	2	<b>612</b>	<b>612</b>	<b>612</b>	<b>612,0</b>	<b>612</b>	<b>612</b>
	3	<b>615</b>	<b>615</b>	626	622,3	642	<b>615</b>
	4	<b>626</b>	<b>626</b>	627	627,6	633	633
	5	<b>609</b>	<b>609</b>	<b>609</b>	<b>609,0</b>	<b>609</b>	<b>609</b>
10	2	774	-	703	<b>687,0</b>	739	717
	3	<b>638</b>	<b>638</b>	676	645,5	681	685
	4	<b>738</b>	<b>738</b>	773	763,8	792	776
	5	<b>706</b>	-	724	713,7	724	724
11	2	789	-	734	<b>708,8</b>	748	753
	3	763	-	785	<b>743,0</b>	765	765
	4	881	-	877	<b>814,3</b>	855	885
	5	695	-	696	692,5	701	<b>691</b>
12	2	625	<b>605</b>	628	608,6	624	<b>605</b>
	3	597	<b>596</b>	597	597,8	600	600
	4	624	-	640	<b>608,0</b>	610	<b>608</b>
	5	602	<b>596</b>	597	598,7	600	600
13	2	3523	-	2775	<b>2714,0</b>	2770	2761
	3	<b>2574</b>	2754	2696	2581,6	2639	2606
	4	<b>2673</b>	-	2743	2686,0	2726	2679
	5	2807	-	2737	<b>2730,9</b>	2735	2735
14	2	1459	-	1266	<b>1242,5</b>	1258	1259
	3	1211	-	1204	<b>1194,2</b>	1213	1224
	4	<b>1166</b>	<b>1166</b>	1187	1180,2	1195	1215

I	Cl	$BC_{ior}$	$BC_{aze}$	$TS_{gen}$	$ACO_{fue}$	$CGV$	$CGN$
	5	1504	-	1309	1289,1	1315	<b>1288</b>
15	2	1203	-	1135	<b>1101,9</b>	1134	-
	3	1405	-	1183	<b>1178,7</b>	1198	-
	4	1358	-	1372	1362,9	1379	<b>1356</b>
	5	1390	-	1361	1361,3	1370	<b>1344</b>
16	2	<b>682</b>	<b>682</b>	<b>682</b>	682,2	689	688
	3	<b>682</b>	<b>682</b>	<b>682</b>	684,3	693	689
	4	<b>691</b>	<b>691</b>	704	693,8	714	695
	5	<b>682</b>	<b>682</b>	<b>682</b>	<b>682,0</b>	687	687
17	2	866	-	<b>851</b>	854,6	<b>851</b>	<b>851</b>
	3	850	-	<b>842</b>	844,5	<b>842</b>	<b>842</b>
	4	853	-	845	844,0	845	<b>842</b>
	5	845	-	<b>842</b>	843,8	<b>842</b>	<b>842</b>

A Tabela 6.5 apresenta os resultados obtidos pelo  $BC_{aze}$  e pelas heurísticas propostas para as instâncias das Classes 2-5 no caso irrestrito. As colunas 1 e 2 apresentam o número da instância e a classe a qual pertence, respectivamente. A coluna 3 apresenta os resultados do algoritmo da literatura, e as colunas 4 e 5 fazem o mesmo para o  $CGV$  e  $CGN$ . Essas 5 colunas são usadas novamente a partir da coluna 6 até a coluna 10, para representar a continuação das instâncias analisadas. Das 76 instâncias consideradas, o  $BC_{aze}$  detém o ótimo de 35 delas. Desses resultados ótimos, o  $CGV$  consegue se igualar a 14, enquanto o  $CGN$ , se iguala a 24. Dentre aqueles valores ótimos que os algoritmos propostos não conseguiram alcançar, a distância percentual média do  $CGV$  foi de 2,67% e a do  $CGN$  foi de 1,43%. Considerando todas as instâncias resolvidas por esses dois algoritmos, o  $CGN$  obteve, na média, resultados 1,27% melhores.

Tabela 6.5: Comparação das instâncias das Classes 2-5 com as *Características Ti* no caso irrestrito.

I	Cl	$BC_{aze}$	$CGV$	$CGN$	I	Cl	$BC_{aze}$	$CGV$	$CGN$
1	2	<b>273</b>	316	289	11	2	-	<b>728</b>	<b>728</b>
	3	<b>279</b>	280	280		3	-	734	<b>733</b>
	4	<b>277</b>	290	<b>277</b>		4	-	834	<b>778</b>
	5	-	<b>279</b>	<b>279</b>		5	-	696	<b>683</b>
2	2	<b>329</b>	336	336	12	2	<b>596</b>	619	610
	3	<b>347</b>	<b>347</b>	<b>347</b>		3	<b>596</b>	597	<b>596</b>
	4	<b>329</b>	336	336		4	-	<b>602</b>	604
	5	<b>329</b>	<b>329</b>	<b>329</b>		5	-	597	<b>596</b>
3	2	<b>381</b>	393	386	13	2	-	2768	<b>2707</b>
	3	<b>387</b>	399	<b>387</b>		3	-	2602	<b>2548</b>
	4	-	<b>374</b>	<b>374</b>		4	-	2717	<b>2686</b>
	5	<b>369</b>	<b>369</b>	<b>369</b>		5	-	2679	<b>2632</b>
4	2	<b>423</b>	<b>432</b>	<b>423</b>	14	2	-	<b>1230</b>	<b>1230</b>
	3	<b>423</b>	438	<b>423</b>		3	-	<b>1206</b>	<b>1206</b>

I	CI	$BC_{aze}$	CGV	CGN	I	CI	$BC_{aze}$	CGV	CGN
	4	<b><u>438</u></b>	<b>438</b>	<b>438</b>	4	-	1177	<b>1173</b>	
	5	<b><u>423</u></b>	<b>423</b>	<b>423</b>	5	-	<b>1288</b>	<b>1288</b>	
5	2	<b><u>367</u></b>	371	<b>367</b>	15	2	-	<b>1105</b>	-
	3	<b><u>373</u></b>	385	<b>373</b>		3	-	<b>1191</b>	<b>1191</b>
	4	<b><u>377</u></b>	<b>377</b>	380		4	-	1366	<b>1336</b>
	5	<b><u>389</u></b>	<b>389</b>	<b>389</b>		5	-	1355	<b>1336</b>
6	2	<b><u>488</u></b>	489	489	16	2	<b><u>682</u></b>	<b>682</b>	<b>682</b>
	3	<b><u>489</u></b>	500	492		3	<b><u>682</u></b>	<b>682</b>	<b>682</b>
	4	<b><u>489</u></b>	497	490		4	<b><u>687</u></b>	693	689
	5	<b><u>488</u></b>	<b>488</b>	<b>488</b>		5	<b><u>682</u></b>	<b>682</b>	<b>682</b>
7	2	-	744	<b>715</b>	17	2	-	<b>851</b>	<b>851</b>
	3	<b><u>685</u></b>	695	<b>685</b>		3	-	<b>842</b>	<b>842</b>
	4	-	<b>722</b>	<b>722</b>		4	-	<b>842</b>	<b>842</b>
	5	<b><u>742</u></b>	<b>742</b>	<b>742</b>		5	-	<b>842</b>	<b>842</b>
8	2	<b><u>680</u></b>	708	<b>680</b>	18	2	-	1061	<b>1047</b>
	3	-	762	<b>729</b>		3	-	<b>1109</b>	-
	4	-	722	<b>711</b>		4	-	1158	<b>1157</b>
	5	-	<b>721</b>	<b>721</b>		5	-	985	<b>978</b>
9	2	<b><u>609</u></b>	622	<b>609</b>	19	2	-	<b>806</b>	824
	3	<b><u>609</u></b>	626	<b>609</b>		3	-	<b>806</b>	823
	4	<b><u>626</u></b>	635	<b>626</b>		4	-	833	<b>786</b>
	5	<b><u>609</u></b>	<b>609</b>	<b>609</b>		5	-	734	<b>728</b>
10	2	-	726	<b>681</b>					
	3	-	677	<b>675</b>					
	4	-	787	<b>737</b>					
	5	-	<b>710</b>	713					

### 6.4.2 Resultados para as *Características Tr*

As tabelas 6.6, 6.7 e 6.8 apresentam os resultados obtidos pelos algoritmos CGV, CGN, BC, e por aqueles propostos na literatura que respeitam as *Características Tr*. Os algoritmos da literatura considerados nas comparações são: a Busca Tabu de Gendreau et al. [2008] ( $TS_{gen}$ ), a Busca Tabu Guiada de Zachariadis et al. [2009] ( $GTS_{zac}$ ), a Otimização de Colônia de Formigas de Fuellerer et al. [2009] ( $ACO_{fue}$ ), o Simulated Annealing de Leung et al. [2010] ( $SA_{leu}$ ), a Busca Tabu Guiada Estendida de Leung et al. [2011] ( $EGTS_{leu}$ ) e o GRASP x ELS de Duhamel et al. [2011] ( $GELS_{duh}$ ). Nas três tabelas apresentadas, os melhores resultados de cada instância estão destacados em negrito, e aqueles sublinhados identificam os ótimos encontrados pelo BC.

A Tabela 6.6 apresenta os valores das soluções obtidas pelos algoritmos da literatura e pelos CGV, CGN e BC, para as instâncias da Classe 1. A primeira coluna indica qual o número da instância associada. As colunas 2-7 apresentam os resultados obtidos pelos algoritmos da literatura. A coluna 8 agrupa os resultados do CGV e CGN, enquanto a coluna 9 exibe os resultados do BC. Das 17 instâncias analisadas, o método BC alcançou

a solução ótima em 14, e o **CGV/CGN** conseguiu obter os melhores resultados em 10. Das instâncias que o **CGV/CGN** não alcançou as melhores soluções, a distância percentual média dos resultados obtidos daqueles melhores foi de 1,53%.

**Tabela 6.6:** Comparação das instâncias da Classe 1 com as *Características Tr*.

<b>I</b>	<i>TS<sub>gen</sub></i>	<i>GTS<sub>zac</sub></i>	<i>ACO<sub>fue</sub></i>	<i>SA<sub>leu</sub></i>	<i>EGTS<sub>leu</sub></i>	<i>GELS<sub>duh</sub></i>	<i>CGV/CGN</i>	<i>BC</i>
1	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>	<b>278,73</b>
2	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>
3	359,77	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	370,80	<b>358,40</b>
4	<b>430,88</b>	<b>430,88</b>	<b>430,88</b>	<b>430,89</b>	<b>430,88</b>	<b>430,88</b>	<b>430,89</b>	<b>430,89</b>
5	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>
6	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>
7	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>
8	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>	<b>568,56</b>
9	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>
10	538,79	<b>535,80</b>	<b>535,80</b>	<b>535,80</b>	<b>535,74</b>	<b>535,80</b>	539,69	<b>535,80</b>
11	<b>505,01</b>	<b>505,01</b>	<b>505,01</b>	<b>505,01</b>	<b>505,01</b>	<b>505,01</b>	511,64	<b>505,01</b>
12	610,57	<b>610,00</b>	<b>610,00</b>	<b>610,00</b>	<b>610,00</b>	<b>610,00</b>	614,94	-
13	<b>2006,34</b>	<b>2006,34</b>	<b>2006,34</b>	<b>2006,34</b>	<b>2006,34</b>	<b>2006,34</b>	2006,68	<b>2006,34</b>
14	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	859,19	<b>837,67</b>
15	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	<b>837,67</b>	855,00	<b>837,67</b>
16	<b>698,61</b>	<b>698,61</b>	<b>698,61</b>	<b>698,61</b>	<b>698,61</b>	<b>698,61</b>	<b>698,61</b>	702,11
17	862,62	863,27	<b>861,79</b>	<b>861,79</b>	<b>861,79</b>	<b>861,79</b>	<b>861,79</b>	-

A Tabela 6.7 apresenta os resultados obtidos pelos mesmos algoritmos usados na tabela anterior, com exceção do *GELS<sub>duh</sub>*, para as instâncias das Classes 2-5 no caso sequencial. As colunas 1 e 2 representam o número da instância e a classe a qual pertence. As colunas 3-7 apresentam os resultados obtidos pelos algoritmos da literatura, enquanto as colunas 7-9 apresentam os resultados do **CGV**, **CGN** e **BC**. Das 68 instâncias apresentadas, o **BC** conseguiu atestar a otimalidade de 35 delas, o **CGV** obteve 12 melhores resultados e o **CGN** obteve 20. Entre as instâncias que as heurísticas propostas não conseguiram o melhor resultado, a distância percentual média entre os valores obtidos e os melhores da literatura são de 3,25% para o **CGV** e 1,89% para o **CGN**.

**Tabela 6.7:** Comparação das instâncias das Classes 2-5 com as *Características Tr* no caso sequencial.

<b>I</b>	<b>Cl</b>	<i>TS<sub>gen</sub></i>	<i>GTS<sub>zac</sub></i>	<i>ACO<sub>fue</sub></i>	<i>SA<sub>leu</sub></i>	<i>EGTS<sub>leu</sub></i>	<i>CGV</i>	<i>CGN</i>	<i>BC</i>
1	2	301,45	319,86	291,49	300,08	317,62	320,00	<b>290,84</b>	<b>290,84</b>
	3	313,91	314,33	306,03	307,53	313,30	315,99	321,39	<b>284,52</b>
	4	296,75	296,75	296,96	296,75	296,75	296,75	<b>294,25</b>	<b>294,25</b>
	5	284,23	285,93	285,93	285,93	285,93	285,93	285,93	<b>278,73</b>
2	2	<b>347,73</b>	<b>347,73</b>	<b>347,73</b>	<b>347,73</b>	<b>347,73</b>	354,01	354,01	<b>347,73</b>
	3	356,24	356,24	356,44	356,24	356,24	356,24	356,24	<b>352,16</b>
	4	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>	<b>342,00</b>
	5	<b>334,96</b>	340,88	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>
3	2	411,24	414,39	403,93	403,93	403,93	414,39	403,93	<b>373,43</b>
	3	<b>394,72</b>	413,63	397,58	413,63	413,63	417,97	<b>394,72</b>	<b>394,72</b>

I	CI	$TS_{gen}$	$GTS_{zac}$	$ACO_{fue}$	$SA_{leu}$	$EGTS_{leu}$	$CGV$	$CGN$	$BC$
	4	376,83	383,11	372,56	382,81	375,60	375,60	382,85	<b>370,82</b>
	5	<b>358,40</b>	362,27	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	365,34	376,18	<b>358,40</b>
4	2	<b>440,94</b>	451,98	<b>440,94</b>	451,98	<b>440,94</b>	451,98	441,97	<b>440,94</b>
	3	446,61	448,24	445,25	446,61	446,61	446,61	441,97	<b>440,68</b>
	4	455,25	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	454,55	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>
	5	<b>430,88</b>	<b>430,88</b>	<b>430,89</b>	<b>430,89</b>	<b>430,89</b>	<b>430,89</b>	<b>430,89</b>	<b>430,89</b>
5	2	390,62	407,45	<b>388,72</b>	<b>388,72</b>	<b>388,72</b>	393,46	390,62	<b>388,72</b>
	3	383,87	401,09	383,15	397,08	389,42	398,77	<b>381,69</b>	<b>381,69</b>
	4	386,47	399,65	385,70	389,28	397,00	408,16	<b>383,88</b>	<b>383,88</b>
	5	<b>375,28</b>	377,26	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>
6	2	<b>499,08</b>	<b>499,08</b>	499,68	<b>499,08</b>	503,29	<b>499,08</b>	<b>499,08</b>	<b>499,08</b>
	3	<b>504,68</b>	509,65	506,45	<b>504,68</b>	<b>504,68</b>	<b>504,68</b>	<b>504,68</b>	<b>504,68</b>
	4	511,52	515,60	501,04	515,05	505,18	512,14	498,65	<b>498,32</b>
	5	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	<b>495,85</b>	502,75	500,59	<b>495,85</b>
7	2	751,15	764,45	734,65	762,26	765,52	767,39	756,29	<b>713,89</b>
	3	<b>702,59</b>	712,57	715,08	709,72	709,72	709,72	713,94	<b>702,59</b>
	4	732,54	723,78	703,64	713,28	713,82	722,49	707,26	<b>703,49</b>
	5	701,31	694,54	669,17	694,54	673,11	702,26	<b>661,22</b>	669,53
8	2	730,87	730,87	727,40	730,87	730,87	731,32	731,32	<b>705,90</b>
	3	771,29	749,70	741,12	752,42	752,42	772,68	745,32	<b>740,24</b>
	4	723,55	727,58	735,06	736,73	727,42	736,73	708,23	<b>697,92</b>
	5	665,20	654,74	649,73	<b>646,46</b>	654,74	656,54	646,73	646,60
9	2	625,13	611,49	611,49	611,49	611,49	611,49	611,49	<b>607,65</b>
	3	638,31	644,54	622,15	640,63	640,63	640,63	613,90	<b>607,65</b>
	4	<b>625,13</b>	<b>625,13</b>	625,46	<b>625,13</b>	<b>625,13</b>	632,14	632,14	-
	5	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>	<b>607,65</b>
10	2	715,51	740,43	<b>700,20</b>	721,77	740,00	788,26	795,48	740,42
	3	660,27	671,24	636,80	658,49	663,40	673,39	674,96	<b>616,32</b>
	4	769,73	770,82	763,98	798,20	803,35	806,55	804,60	<b>728,05</b>
	5	714,08	728,95	<b>705,62</b>	712,43	720,03	719,04	707,68	723,49
11	2	754,62	748,96	722,44	742,88	748,96	763,87	768,90	<b>697,71</b>
	3	789,95	783,92	730,30	782,10	779,34	779,73	771,27	<b>717,37</b>
	4	899,46	868,01	832,91	852,87	861,34	870,87	901,71	<b>777,79</b>
	5	<b>649,75</b>	671,71	660,45	663,70	661,68	666,80	694,00	-
12	2	641,84	638,06	620,47	629,82	629,82	638,06	<b>619,63</b>	-
	3	610,57	610,57	612,21	<b>610,00</b>	614,59	615,41	615,41	-
	4	664,76	655,60	623,71	624,01	630,40	<b>623,21</b>	623,79	-
	5	<b>610,23</b>	<b>610,23</b>	611,87	<b>610,23</b>	614,59	614,94	614,94	641,15
13	2	2836,79	2778,28	2669,39	2763,95	2771,19	2783,82	-	<b>2525,91</b>
	3	2625,82	2660,74	<b>2500,55</b>	2567,96	2580,19	2629,87	-	2551,05
	4	2743,17	2748,57	2703,48	2743,53	2731,91	2743,53	2725,53	<b>2609,36</b>
	5	2542,34	2529,77	2520,42	<b>2490,87</b>	2508,22	2557,13	2511,29	-
14	2	1195,39	1185,52	1139,70	1140,40	1145,94	1192,28	-	<b>1121,33</b>
	3	1156,33	1139,74	<b>1101,96</b>	1118,46	1137,50	1157,35	-	1210,27
	4	1058,98	1036,37	<b>1000,76</b>	1009,00	1014,48	1155,41	-	1029,85
	5	995,25	1009,49	<b>959,51</b>	976,53	1002,22	1055,76	-	-
15	2	1143,73	1120,00	<b>1116,51</b>	1116,85	1120,62	1159,35	-	1125,69
	3	1209,60	1188,63	1195,89	<b>1187,59</b>	1189,09	1259,48	-	-
	4	1311,09	1329,39	1292,35	1289,67	1288,84	1316,93	-	<b>1278,22</b>

I	Cl	$TS_{gen}$	$GTS_{zac}$	$ACO_{fue}$	$SA_{leu}$	$EGTS_{leu}$	$CGV$	$CGN$	$BC$
	5	1299,14	1298,80	1287,38	<b>1265,14</b>	1294,20	1305,83	-	-
16	2	<b>698,61</b>	704,58	<b>698,60</b>	<b>698,61</b>	700,49	704,80	703,57	<b>698,61</b>
	3	<b>698,61</b>	708,83	700,10	706,61	703,57	708,83	<b>698,61</b>	705,06
	4	723,58	718,21	<b>709,57</b>	<b>709,27</b>	712,30	731,13	711,43	776,00
	5	<b>698,61</b>	<b>698,61</b>	<b>698,60</b>	<b>698,61</b>	<b>698,61</b>	703,35	703,35	709,59
17	2	874,34	<b>870,86</b>	874,02	<b>870,86</b>	<b>870,86</b>	<b>870,86</b>	<b>870,86</b>	-
	3	862,62	<b>861,79</b>	863,06	<b>861,79</b>	873,12	<b>861,79</b>	<b>861,79</b>	-
	4	866,42	864,58	862,62	862,62	862,62	862,62	<b>861,79</b>	-
	5	862,62	863,58	862,53	<b>861,79</b>	873,12	<b>861,79</b>	<b>861,79</b>	-

Na Tabela 6.8, são apresentados os resultados obtidos pelos algoritmos, utilizados na Tabela 6.6, para as instâncias das Classes 2-5 no caso irrestrito. As duas primeiras colunas apresentam o número da instância e a classe a qual pertence, respectivamente. As colunas 3-8 apresentam os resultados dos algoritmos da literatura, e as colunas 9-11 fazem o mesmo para o  $CGV$ ,  $CGN$  e  $BC$ , respectivamente. Das 68 instâncias consideradas, o  $BC$  encontrou o ótimo de 35, o  $CGV$  conseguiu obter 16 melhores resultados e o  $CGN$ , 26. Considerando somente as instâncias que as heurísticas propostas não conseguiram alcançar os melhores valores, a distância percentual da média entre os valores obtidos e os melhores da literatura são de 4,6% e 1,67% para o  $CGV$  e  $CGN$ , respectivamente.

Tabela 6.8: Comparação das instâncias das Classes 2-5 com as *Características Tr* no caso irrestrito.

I	Cl	$TS_{gen}$	$GTS_{zac}$	$ACO_{fue}$	$SA_{leu}$	$EGTS_{leu}$	$GELS_{duh}$	$CGV$	$CGN$	$BC$
1	2	284,42	305,92	284,52	285,50	304,79	284,42	317,96	294,27	<b>278,73</b>
	3	306,31	299,70	296,87	299,70	299,70	<b>284,52</b>	<b>284,52</b>	<b>284,52</b>	<b>284,52</b>
	4	296,94	296,75	<b>282,95</b>	296,75	296,75	<b>282,95</b>	296,75	<b>282,95</b>	<b>282,95</b>
	5	<b>278,73</b>	280,60	<b>278,73</b>	<b>278,73</b>	280,60	<b>278,73</b>	285,93	285,93	<b>278,73</b>
2	2	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	342,00	342,00	<b>334,96</b>
	3	<b>352,16</b>	355,65	<b>352,16</b>	353,48	355,65	<b>352,16</b>	353,06	353,06	<b>352,16</b>
	4	342	342,00	342,00	342,00	342,00	<b>334,96</b>	342,00	342,00	<b>334,96</b>
	5	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>	<b>334,96</b>
3	2	387,7	401,81	387,70	387,70	387,70	387,70	400,15	393,54	<b>373,29</b>
	3	394,72	409,17	394,72	398,90	402,98	394,72	407,45	394,72	<b>390,55</b>
	4	368,56	368,56	364,45	364,45	368,56	364,45	374,11	372,23	<b>362,41</b>
	5	<b>358,4</b>	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	<b>358,40</b>	367,26	368,34	<b>358,40</b>
4	2	<b>430,88</b>	440,94	<b>430,88</b>	<b>430,89</b>	438,52	<b>430,88</b>	440,68	<b>430,89</b>	<b>430,89</b>
	3	440,68	446,61	445,49	<b>430,89</b>	446,61	<b>430,88</b>	446,61	<b>430,89</b>	<b>430,89</b>
	4	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>	<b>447,37</b>
	5	<b>430,88</b>	<b>430,88</b>	<b>430,88</b>	<b>430,89</b>	<b>430,89</b>	<b>430,88</b>	<b>430,89</b>	<b>430,89</b>	<b>430,89</b>
5	2	379,94	381,85	<b>375,28</b>	<b>375,28</b>	381,85	<b>375,28</b>	379,94	<b>375,28</b>	<b>375,28</b>
	3	<b>381,69</b>	387,89	<b>381,69</b>	383,52	387,89	<b>381,69</b>	392,90	<b>381,69</b>	<b>381,69</b>
	4	<b>383,87</b>	<b>383,87</b>	<b>383,88</b>	<b>383,88</b>	<b>383,88</b>	<b>383,87</b>	<b>383,88</b>	<b>383,88</b>	<b>383,88</b>
	5	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>	<b>375,28</b>
6	2	498,16	498,16	<b>495,85</b>	498,16	498,16	<b>495,85</b>	498,16	499,08	<b>495,85</b>





## 6.5 Comentários sobre os Resultados

De modo geral, pode-se observar que os algoritmos **CGV** e **CGN** obtiveram bons resultados. Vale ressaltar, que essas heurísticas se baseiam somente nas colunas geradas para o **RLMP** referente a raiz da árvore de **BP** que resolve o **MP**. Por isso, normalmente, os resultados são valores acima daqueles obtidos por outras heurísticas da literatura. Mesmo assim, nas instâncias que os algoritmos propostos foram capazes de resolver, os seus resultados comparados àqueles da literatura, na média, não foram piores em mais de 5%, no caso do **CGV**, e 2%, no caso do **CGN**.

Nos experimentos realizados, pôde ser notado que o **CGN** obteve melhores resultados que o **CGV**, na maioria das vezes. Isso ocorre, porque a verificação da viabilidade dos carregamentos no subproblema de precificação do **CGV**, não somente prejudica o tempo de execução do mesmo, como, também, a qualidade das soluções. Essa perda de qualidade é consequência do descarte de colunas viáveis que são consideradas inviáveis pelas heurísticas de verificação de viabilidade. Esse comportamento do algoritmo resulta em seu término precipitado, pois algumas colunas que ainda seriam geradas, podem ter sido descartadas anteriormente por terem sido consideradas inviáveis. Em contrapartida, o **CGN** tem a verificação da viabilidade realizada também pelo **BB Aproximado**, o qual resulta em um número menor de falso-negativos. Por isso, o valor final da relaxação linear obtido pelo **CGV** é normalmente pior que aquele obtido pelo **CGN** e, conseqüentemente, o mesmo vale para as soluções inteiras dadas pelos dois.

Apesar de bons resultados, ambas heurísticas demandam muito tempo computacional em instâncias ainda pequenas. Isso torna oneroso sua utilização em um algoritmo **BP**, onde a cada nó da árvore de enumeração é resolvido um problema de **CG**. O tempo de execução gasto pelos algoritmos **CGV** e **CGN**, em sua maioria, é oriundo do método de programação dinâmica utilizado para resolver o subproblema de precificação. Observando-se as tabelas do Apêndice **A**, isso se torna mais claro nos resultados do **CGN**, onde os tempos de roteamento são normalmente maiores que os tempos de carregamento. Por outro lado, no caso do **CGV**, há várias instâncias onde o tempo de carregamento é maior que o tempo de roteamento. Contudo, vale lembrar que a análise dos carregamentos é feita no momento da precificação, logo, ainda pode-se considerar que esse tempo é proveniente da programação dinâmica<sup>1</sup>.

O algoritmo **BC** proposto conseguiu atestar a otimalidade, pela primeira vez, de instâncias com as *Características Tr*. Ao todo, foram encontradas as soluções ótimas de 84 instâncias, onde 14 são da Classe 1, 35 são do caso sequencial e 35 do caso irrestrito. O **BC** conseguiu chegar ao final de sua execução para mais 9 instâncias, porém, o algoritmo **BB** para verificar a viabilidade das rotas falhou pelo menos uma vez em cada uma delas,

---

<sup>1</sup>Foi observado que os algoritmos de precificação heurísticos e a resolução dos problemas lineares demandaram muito pouco tempo computacional, por isso, pode-se considerar que o tempo de roteamento é, basicamente, o mesmo que aquele gasto pela programação dinâmica.

ou seja, o tempo máximo de execução do **BB** foi atingido, e ele teve que considerar a rota inviável. Por esse motivo, as soluções dessas instâncias não são, garantidamente, ótimas. Contudo, testes preliminares com esse **BB** apontaram que nos casos onde o algoritmo executa por mais de 7200 segundos, a chance do carregamento ser viável é muito pequena. Além disso, vale observar que, na maioria dos casos, a maior parcela do tempo de execução do **BC** consistiu na avaliação das rotas no procedimento de separação das **IPCs**, apesar da maioria dos cortes gerados serem **CCCs**.

Por fim, pode-se destacar que a resolução das instâncias considerando as *Características Tr* é mais difícil que aquelas com *Características Ti*. Do ponto de vista do roteamento, nas *Características Tr*, o número de combinações de rotas possíveis aumenta muito, pois além das possibilidades disponíveis para as *Características Ti*, existem também os casos onde o número de rotas é menor que o número de veículos e aqueles onde há rotas com apenas um cliente. Do ponto de vista do carregamento, o problema também pode se tornar mais complexo com as *Características Tr*. Nesse caso, o número de rotas tende a diminuir e, conseqüentemente, o número de clientes por rota tende a aumentar. Isso acarreta num aumento no número de itens por carregamento, o que dificulta a verificação de sua viabilidade. A partir dos experimentos realizados, pôde-se constatar que tanto o tempo de carregamento quanto o tempo de roteamento foram maiores quando utilizadas as *Características Tr* ao invés das *Ti*.



# Capítulo 7

## Conclusão e Trabalhos Futuros

Neste trabalho foi abordado o Problema de Roteamento de Veículos Capacitado com Restrições de Carregamento Bidimensional (**2L-CVRP**). Esse problema consiste em minimizar os custos de transporte ao realizar as entregas das demandas aos clientes. No **2L-CVRP**, um fator complicante é inserido ao se considerar que os clientes demandam itens retangulares, os quais devem ser posicionados no compartimento dos veículos sem que haja sobreposição e impedimento entre eles. Dois casos foram avaliados: (a) o irrestrito, onde não há restrição com relação ao posicionamento dos itens, e (b) o sequencial, onde os itens devem ser posicionados de tal modo que a sequencialidade da entrega seja respeitada. Além desses casos, dois conjuntos de *Características* foram considerados: (a)  $T_i$ , que representa o caso onde as distâncias entre os clientes é truncada, o número de rotas deve ser igual ao número de veículos e não podem existir rotas com um único cliente, e (b)  $Tr$ , no qual as distâncias entre os clientes são os valores reais, podem haver menos rotas que o número de veículos e também rotas com apenas um cliente.

Duas novas heurísticas baseadas em **CG**, denominadas **CGV** e **CGN**, foram propostas para a resolução do **2L-CVRP**. Até onde se sabe, essas são as primeiras abordagens hibridizadas de métodos heurísticos e exatos para resolver esse problema. Ambas heurísticas seguem o fluxo de funcionamento de uma **CG** padrão e, quando esse processo termina, o **RLMP** resultante tem as variáveis convertidas para inteiras e um **BB** é executado a fim de encontrar uma solução válida para o **2L-CVRP**. O que diferencia os dois algoritmos propostos diz respeito a verificação da viabilidade dos carregamentos. No **CGV**, a viabilidade é garantida por heurísticas durante a geração de novas colunas, portanto, sua solução final é sempre viável. Por outro lado, o **CGN** só verifica a viabilidade dos carregamentos após conversão das variáveis do **RLMP** para inteiras e a sua resolução pelo **BB**. Se nesse momento, algum carregamento for inviável, sua rota associada é proibida de ser gerada novamente, e o processo de **CG** se repete. Esse algoritmo termina quando todas as rotas tiverem os carregamentos viáveis.

Os experimentos conduzidos mostraram que as heurísticas propostas obtiveram bons

resultados, dado que as soluções obtidas são fruto da resolução da raiz de uma árvore de **BP**. Pôde-se observar que a árvore de viabilidade foi muito eficiente, sendo responsável por evitar que os métodos de verificação fossem chamados em mais de 85% das vezes. O que, em parte, pode ser explicado pelo fato da árvore poder inferir sobre a viabilidade de carregamentos não avaliados pelos algoritmos de avaliação da carga. Além disso, foi verificado que no **CGV** e **CGN**, a resolução do subproblema de precificação é o maior contribuidor para os seus tempos totais de execução. Finalmente, foi constatado que, na média, o **CGN** superou os resultados do **CGV**. A justificativa para esse resultado é o fato do **CGV** verificar a viabilidade dos carregamentos via heurísticas, o que, normalmente, gera resultados falso-negativos e, conseqüentemente, resultam na eliminação de rotas viáveis da solução.

Além das heurísticas propostas, nesta dissertação foi desenvolvido um algoritmo **BC**, baseado naquele mesmo de Iori et al. [2007], para resolver as instâncias com as *Características Tr*, que ainda não haviam sido resolvidas por um método exato. Esse método foi capaz de resolver 84 instâncias à otimalidade, onde 14 delas são referentes à Classe 1, 35 ao caso sequencial e outras 35 ao caso irrestrito. O desempenho do **BC** cai rapidamente a medida que o número de clientes aumenta, por isso, a maior instância resolvida contém apenas 35 clientes.

Com relação à verificação da viabilidade dos carregamentos, três métodos foram utilizados pelos algoritmos de resolução do **2L-CVRP**. O primeiro deles consistiu em uma coleção de heurísticas construtivas reunidas para obter resultado em um tempo computacional muito baixo. O segundo foi um algoritmo **BB**, utilizado para definir de forma exata a viabilidade dos carregamentos. Por último, um **BB** Aproximado foi proposto para tentar obter resultados fiéis àqueles obtidos pelo **BB** exato, com um esforço computacional menor.

Em trabalhos futuros, seria interessante desenvolver regras de dominância específicas para o **2L-CVRP**, as quais levassem em conta os carregamentos dos veículos. Uma outra opção, seria propor um novo método para resolver o subproblema de precificação, o qual conseguisse mensurar a qualidade das rotas a partir dos seus carregamentos, ou que até mesmo desvinculasse o problema de carregamento do problema de roteamento. Com relação ao **CGV**, seu desempenho poderia ser melhorado com o uso de heurísticas mais eficientes para analisar os carregamentos, ou através de considerações bem fundamentadas que permitissem reduzir o número de vezes que as heurísticas fossem aplicadas durante a precificação. No que diz respeito às verificações dos carregamentos, ainda não são conhecidos limites inferiores que levem em conta a sequência dos itens, portanto, o desenvolvimento de um poderia ajudar a diminuir a necessidade do uso de algoritmos de verificação. Por fim, de modo geral, seria interessante desenvolver métodos que precisassem resolver o problema de carregamento o menor número de vezes possível. Para isso, um algoritmo **BCP** poderia ser proposto, onde a viabilidade do carregamento fosse

garantida através da inserção de cortes, e boas rotas fossem garantidas através da geração das colunas.





# Referências Bibliográficas

- Alvarenga, G. B.; Mateus, G. R. & de Tomi, G. (2007). A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, 34(6):1561–1584. Part Special Issue: Odysseus 2003 Second International Workshop on Freight Transportation Logistics.
- Araújo, O. C. B. d. (2006). *Problemas de corte e empacotamento tridimensional e integração com roteamento de veículos*. Tese de doutorado, Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Sistemas, Universidade Estadual de Campinas, Campinas, SP, Brasil.
- Azevedo, B. L. P. (2009). Uma abordagem exata para o problema de roteamento de veículos capacitados com restrições bidimensionais de carregamento. Dissertação de mestrado, Instituto de Computação, Universidade Estadual de Campinas, Campinas, SP, Brasil.
- Azevedo, B. L. P.; Hokama, P. H.; Miyazawa, F. K. & Xavier, E. C. (2009). A branch-and-cut approach for the vehicle routing problem with two-dimensional loading constraints. Em *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional*, pp. 1–12, Porto Seguro, Brasil.
- Balinski, M. L. & Quandt, R. E. (1964). On an integer program for a delivery problem. *Operations Research*, 12(2):300–304.
- Beasley, J. E. & Christofides, N. (1989). An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394.
- Boland, N.; Dethridge, J. & Dumitrescu, I. (2006). Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34(1):58–68.
- Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers and Operations Research*, 39(9):2248–2257.

- Bramel, J. & Simchi-Levi, D. (2001). Set-covering-based algorithms for the capacitated VRP. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 85–108. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Burke, E. K.; Kendall, G. & Whitwell, G. (2004). A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, 52(4):655–671.
- Chazelle, B. (1983). The bottomn-left bin-packing heuristic: An efficient implementation. *IEEE Transactions on Computers*, 32(8):697–707.
- Clarke, G. & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12(4):568–581.
- Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management Science*, 6(1):80–91.
- Dell’Amico, M. & Martello, S. (1995). Optimal scheduling of tasks on identical parallel processors. *INFORMS Journal on Computing*, 7(2):191–200.
- Desaulniers, G.; Desrosiers, J.; Erdmann, A.; Solomon, M. M. & Soumis, F. (2001). VRP with pickup and delivery. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 225–242. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Desrochers, M. (1988). An algorithm for the shortest path problem with resource constraints. Relatório técnico, Group for Research in Decision Analysis.
- Doerner, K. F.; Fuellerer, G.; Hartl, R. F.; Gronalt, M. & Iori, M. (2007). Metaheuristics for the vehicle routing problem with loading constraints. *Networks*, 49(4):294–307.
- Dowsland, K. A. & Dowsland, W. B. (1992). Packing problems. *European Journal of Operational Research*, 56(1):2–14.
- Dror, M. (1994). Note on the complexity of the shortest path models for column generation in vrptw. *Operations Research*, 42(5):977–978.
- Duhamel, C.; Lacomme, P.; Quilliot, A. & Toussaint, H. (2011). A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers and Operations Research*, 38(3):617–640.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159.

- Feillet, D.; Dejax, P.; Gendreau, M. & Gueguen, C. (2004). An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229.
- Fuellerer, G.; Doerner, K. F.; Hartl, R. F. & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, 36(3):655–673.
- Fuellerer, G.; Doerner, K. F.; Hartl, R. F. & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, 201(3):751–759.
- Fukasawa, R.; Longo, H.; Lysgaard, J.; de Aragão, M. P.; Reis, M. L.; Uchoa, E. & Werneck, R. F. F. (2006). Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511.
- Gendreau, M.; Iori, M.; Laporte, G. & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, 40(3):342–350.
- Gendreau, M.; Iori, M.; Laporte, G. & Martello, S. (2008). A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. *Networks*, 51(1):4–18.
- Gendreau, M.; Laporte, G. & Potvin, J.-Y. (2001). Metaheuristics for the capacitated VRP. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 129–154. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Gilmore, P. C. & Gomory, R. E. (1961). A linear programming approach to the cutting-stock problem. *Operations Research*, 9(6):849–859.
- Gilmore, P. C. & Gomory, R. E. (1963). A linear programming approach to the cutting stock problem - part ii. *Operations Research*, 11(6):863–888.
- Gilmore, P. C. & Gomory, R. E. (1965). Multistage cutting stock problems of two and more dimensions. *Operations Research*, 13(1):94–120.
- Ibaraki, T. (1987). Chapter 8 design and analysis of approximate algorithms. *Annals of Operations Research*, 11:441–510.
- Iori, M. (2004). *Metaheuristic algorithms for combinatorial optimization problems*. Tese de doutorado, DEIS - Department of Electronics, Computer Science and Systems, University of Bologna, Bologna, Italy.

- Iori, M. & Martello, S. (2010). Routing problems with loading constraints. *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research*, 18(1):4–27.
- Iori, M.; Salazar-González, J.-J. & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation Science*, 41(2):253–264.
- Kantorovich, L. V. (1960). Mathematical methods of organizing and planning production. *Management Science*, 6(4):366–422.
- Laporte, G. & Semet, F. (2001). Classical heuristics for the capacitated VRP. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 109–128. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Leung, S. C. H.; Zheng, J.; Zhang, D. & Zhou, X. (2010). Simulated annealing for the vehicle routing problem with two-dimensional loading constraints. *Flexible Services and Manufacturing Journal*, 22:61–82.
- Leung, S. C. H.; Zhou, X.; Zhang, D. & Zheng, J. (2011). Extended guided tabu search and a new packing algorithm for the two-dimensional loading vehicle routing problem. *Computers and Operations Research*, 38(1):205–215.
- Lodi, A.; Martello, S. & Monaci, M. (2002). Two-dimensional packing problems: A survey. *European Journal of Operational Research*, 141(2):241–252.
- Lodi, A.; Martello, S. & Vigo, D. (1999). Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. *INFORMS Journal on Computing*, 11(4):345–357.
- Lopes, R. P. (2012). Algoritmos exatos e heurísticos para problemas seletivos de roteamento de veículos com restrições de cobertura. Dissertação de mestrado, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brasil.
- Lourenço, H. R.; Martin, O. C. & Stutzle, T. (2010). Iterated local search: Framework and applications. Em Gendreau, M. & Potvin, J.-Y., editores, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research and Management Science*, capítulo 12, pp. 363–397. Springer, 2ª edição.
- Martello, S.; Pisinger, D. & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations Research*, 48(2):256–267.

- Martello, S. & Toth, P. (1990). Lower bounds and reduction procedures for the bin packing problem. *Discrete Applied Mathematics*, 28(1):59–70.
- Martello, S. & Vigo, D. (1998). Exact solution of the two-dimensional finite bin packing problem. *Management Science*, 44(3):388–399.
- Miller, C. E.; Tucker, A. W. & Zemlin, R. A. (1960). Integer programming formulation of traveling salesman problems. *Journal of the ACM*, 7(4):326–329.
- Moura, A. (2008). A multi-objective genetic algorithm for the vehicle routing with time windows and loading problem. Em Bortfeldt, A.; Homberger, J.; Kopfer, H.; Pankratz, G. & Strangmeier, R., editores, *Intelligent Decision Support*, pp. 187–201. Gabler.
- Moura, A. & Oliveira, J. (2009). An integrated approach to the vehicle routing and container loading problems. *OR Spectrum*, 31(4):775–800.
- Naddef, D. & Rinaldi, G. (2001). Branch-and-cut algorithms for the capacitated VRP. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 53–84. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Pinto, T.; Alves, C. & de Carvalho, J. V. (2011). An overview on recent approaches for vehicle routing problems with loading constraints. Em *Proceedings of the 10o Congreso Galego de Estatística e Investigación de Operacións*, Pontevedra, Spain.
- Righini, G. & Salani, M. (2006). Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273.
- Ruan, Q.; Zhang, Z.; Miao, L. & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers and Operations Research*, 40(6):1579–1589.
- Scheithauer, G. (1997). Equivalence and dominance for problems of optimal packing of rectangles. *Ricerca Operativa*, 27(83):3–34.
- Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.
- Tarantilis, C.; Zachariadis, E. & Kiranoudis, C. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *Intelligent Transportation Systems, IEEE Transactions on*, 10(2):255–271.

- Toth, P. & Vigo, D. (2001a). Branch-and-bound algorithms for the capacitated VRP. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 29–51. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Toth, P. & Vigo, D., editores (2001b). *The vehicle routing problem*. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Toth, P. & Vigo, D. (2001c). VRP with backhauls. Em Toth, P. & Vigo, D., editores, *The vehicle routing problem*, pp. 195–224. Society for Industrial and Applied Mathematics monographs on discrete mathematics and applications, Philadelphia, PA, USA.
- Wang, F.; Tao, Y. & Shi, N. (2009). A survey on vehicle routing problem with loading constraints. Em *Proceedings of the 2009 International Joint Conference on Computational Sciences and Optimization*, volume 02 of *CSO '09*, pp. 602–606, Washington, DC, USA. IEEE Computer Society.
- Wäscher, G.; Haußner, H. & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130.
- Zachariadis, E. E.; Tarantilis, C. D. & Kiranoudis, C. T. (2009). A guided tabu search for the vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, 195(3):729–743.

# Apêndice A

## Resultados Detalhados

A seguir são apresentados os resultados detalhados para os algoritmos desenvolvidos nesta dissertação. A Seção A.1 se refere aos resultados obtidos pelo CGV e CGN para as instâncias com *Características Ti*, e a Seção A.2 faz o mesmo para os algoritmos CGV, CGN e BC, considerando as instâncias com *Características Tr*.

### A.1 Resultados para as Instâncias com *Características Ti*

As Tabelas A.1-A.5 apresentam os resultados dos algoritmos CGV e CGN para as instâncias com *Características Ti*. As colunas em comum dessas Tabelas são explicadas a seguir. A coluna **I** indica o número da instância analisada. A coluna  $\#it_{CG}$  apresenta o número de iterações realizadas pela CG (laço das linhas 2-6 do Algoritmo 7, ou das linhas 3-7 do Algoritmo 8). As colunas RL e LS exibem os valores da relaxação linear do RLMP e o valor da solução inteira (limite superior) obtidos ao término da execução do algoritmo. A coluna  $gap_{RL}(\%)$  apresenta o *gap* percentual entre o limite superior e o valor da relaxação linear, calculado pela equação  $100 \times (LS - RL)/LS$ . As colunas  $lit^*$  e  $gap_{lit^*}(\%)$  apresentam o valor da melhor solução obtida na literatura para a instância em questão e o *gap* percentual entre o limite superior e esse melhor valor, através da equação  $100 \times (LS - lit^*)/LS$ . Caso a solução obtida pelo algoritmo proposto seja melhor ou igual àquela da literatura, o  $gap_{lit^*}(\%)$  associado é destacado em negrito. A coluna  $T_{tot}(s)$  relata o tempo total de execução do algoritmo testado, em segundos. Com relação às Tabelas A.2-A.5, há três colunas em comum: **Cl**, que apresenta a Classe da instância relacionada;  $T_{car}(s)$ , que indica qual o tempo, em segundos, gasto para analisar os carregamentos dos veículos; e  $T_{rot}(s)$ , que apresenta o tempo, em segundos, usado pelos algoritmos para realizar as operações de roteamento.

As colunas  $\#verif$  e  $arv(\%)$  são exclusivas dos resultados do CGV, nas Tabelas A.2

e A.3. A primeira representa o número de carregamentos que foram analisados, enquanto a última apresenta a porcentagem desses carregamentos que conseguiram resposta sobre suas viabilidades apenas verificando a árvore de viabilidade. Por sua vez, as colunas  $\#it_{EL}$  e  $\#Cols_{rej}$  estão presentes somente nas Tabelas A.4 e A.5, referentes aos resultados do CGN. A primeira dessas colunas indica o número de iterações executadas pelo laço das linhas 2-20 do Algoritmo 8, e a segunda apresenta o número de colunas (rotas) que foram rejeitadas após o teste de viabilidade de seus carregamentos.

**Tabela A.1:** Resultados detalhados do CGV/CGN para a Classe 1 com as *Características*  $T_i$ .

I	$\#it_{CG}$	RL	LS	$gap_{RL}(\%)$	$lit^*$	$gap_{lit^*}(\%)$	$T_{tot}(s)$
1	18	265,33	273	2,81	273	<b>0,00</b>	1,51
2	3	321,33	329	2,33	329	<b>0,00</b>	0,09
3	33	344,50	362	4,83	351	3,04	1,02
4	9	423,00	423	0,00	423	<b>0,00</b>	0,05
5	34	366,25	367	0,20	367	<b>0,00</b>	0,39
6	40	472,50	488	3,18	488	<b>0,00</b>	0,52
7	144	558,00	558	0,00	558	<b>0,00</b>	0,46
8	58	653,00	657	0,61	657	<b>0,00</b>	0,15
9	25	609,00	609	0,00	609	<b>0,00</b>	1,86
10	284	475,11	527	9,85	524	0,57	643,07
11	276	494,09	500	1,18	500	<b>0,00</b>	60,79
12	31	589,25	596	1,13	596	<b>0,00</b>	5,22
13	835	1971,86	1991	0,96	1991	<b>0,00</b>	3600,10
14	281	825,18	845	2,35	823	2,60	3600,26
15	515	898,86	907	0,90	907	<b>0,00</b>	3600,17
16	36	679,05	688	1,30	682	0,87	0,95
17	50	841,83	842	0,02	842	<b>0,00</b>	6,62
18	1740	718,00	718	0,00	-	-	14400,09
19	359	528,78	533	0,79	-	-	14400,34
20	2395	238,00	238	0,00	-	-	14400,08
21	2187	663,54	680	2,42	-	-	3601,08
22	1402	716,86	745	3,78	-	-	3601,77
23	469	832,08	852	2,34	-	-	3600,49
24	231	1014,24	1032	1,72	-	-	3600,45
25	3078	798,42	808	1,19	-	-	3602,21
26	2013	795,50	796	0,06	-	-	3600,17
27	805	1061,55	1104	3,85	-	-	3600,99
28	4120	1032,00	1032	0,00	-	-	3600,69
29	4108	1178,33	1213	2,86	-	-	3602,93
30	2544	1021,51	1027	0,53	-	-	3600,23
31	1071	1452,01	1623	10,54	-	-	3663,32
32	3642	1300,05	1339	2,91	-	-	3667,53
33	3589	1304,57	1317	0,94	-	-	3604,13
34	3489	586,86	639	8,16	-	-	4102,38
35	2611	860,64	869	0,96	-	-	3603,16
36	14140	546,51	547	0,09	-	-	3605,24
<b>Média</b>	-	-	-	2,08	-	0,42	-



**Tabela A.2:** Resultados detalhados do **CGV** para as Classes 2-5 com as *Características  $T_i$*  no caso sequencial.

I	Cl	$\#it_{CG}$	$\#\text{verif}$	arv(%)	RL	LS	$gap_{RL}(\%)$	$lit^*$	$gap_{lit^*}(\%)$	$T_{car}(s)$	$T_{rot}(s)$	$T_{tot}(s)$
1	2	26	561889	87,51	297,20	313	5,05	285,0	8,95	9,06	9,14	18,20
	3	28	424858	87,14	299,00	310	3,55	280,0	9,68	8,50	4,33	12,83
	4	6	6574	68,92	290,00	290	0,00	288,0	0,69	0,26	0,00	0,26
	5	6	1770	62,43	279,00	279	0,00	279,0	<b>0,00</b>	0,06	0,01	0,07
2	2	1	2289	43,34	337,00	347	2,88	342,0	1,44	0,06	0,13	0,19
	3	1	1812	42,33	341,00	350	2,57	347,0	0,86	0,09	0,11	0,20
	4	6	3570	73,75	321,33	336	4,37	336,0	<b>0,00</b>	0,05	0,23	0,28
	5	3	2050	55,17	321,33	329	2,33	329,0	<b>0,00</b>	0,04	0,11	0,15
3	2	17	26230	68,29	399,50	407	1,84	389,0	4,42	0,76	0,24	1,00
	3	26	145256	84,64	409,00	409	0,00	387,0	5,38	3,68	0,59	4,27
	4	21	20034	75,12	374,25	377	0,73	374,0	0,80	0,45	0,08	0,53
	5	28	20622	82,96	365,00	369	1,08	369,0	<b>0,00</b>	0,27	0,11	0,38
4	2	13	35080	71,65	445,00	445	0,00	434,0	2,47	0,90	0,11	1,01
	3	12	22092	62,56	433,50	438	1,03	432,0	1,37	0,79	0,19	0,98
	4	6	17628	58,69	438,00	438	0,00	438,0	<b>0,00</b>	1,03	0,12	1,15
	5	16	7721	75,51	423,00	423	0,00	423,0	<b>0,00</b>	0,13	0,03	0,16
5	2	74	1834942	82,85	385,00	385	0,00	380,0	1,30	40,78	50,96	91,74
	3	93	4392756	91,63	389,00	389	0,00	373,0	4,11	60,68	103,73	164,41
	4	54	808617	85,64	384,25	400	3,94	377,0	5,75	15,53	11,63	27,16
	5	52	53704	86,69	382,00	389	1,80	389,0	<b>0,00</b>	0,70	0,21	0,91
6	2	31	104666	84,63	484,29	491	1,37	491,0	<b>0,00</b>	1,52	0,71	2,23
	3	29	115644	86,61	496,00	496	0,00	496,0	<b>0,00</b>	1,99	0,58	2,57
	4	30	84509	84,63	500,67	503	0,46	489,0	2,78	2,25	0,47	2,72
	5	19	77245	87,98	472,50	494	4,35	488,0	1,21	0,95	0,44	1,39
7	2	39	136709	81,26	741,00	757	2,11	724,0	4,36	3,08	0,39	3,47
	3	48	85281	79,90	698,00	698	0,00	698,0	<b>0,00</b>	2,25	0,16	2,41
	4	43	99136	85,32	735,67	736	0,05	714,0	2,99	2,40	0,22	2,62
	5	20	23518	74,71	740,50	746	0,74	742,0	0,54	0,92	0,09	1,01
8	2	31	70242	73,74	717,50	720	0,35	720,0	<b>0,00</b>	1,80	0,24	2,04
	3	48	120322	77,24	762,00	763	0,13	730,0	4,33	4,02	0,41	4,43

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	54	108414	84,56	719,25	726	0,93	701,0	3,44	2,79	0,26	3,05
	5	40	41215	80,60	728,25	736	1,05	721,0	2,04	1,15	0,10	1,25
9	2	10	25901	74,96	612,00	612	0,00	612,0	<b>0,00</b>	0,50	0,13	0,63
	3	19	58666	68,80	639,67	642	0,36	615,0	4,21	2,95	0,39	3,34
	4	24	58595	71,23	616,75	633	2,57	626,0	1,11	2,05	0,50	2,55
	5	21	40091	72,88	609,00	609	0,00	609,0	<b>0,00</b>	1,37	0,24	1,61
10	2	119	5786719	93,54	711,02	739	3,79	687,0	7,04	51,43	56,68	108,11
	3	273	8389409	92,95	644,26	681	5,39	638,0	6,31	101,46	69,82	171,28
	4	114	2345858	94,09	760,50	792	3,98	738,0	6,82	26,38	14,32	40,70
	5	82	460762	80,52	700,80	724	3,20	706,0	2,49	21,03	2,04	23,07
11	2	178	2796351	91,39	696,02	748	6,95	708,8	5,24	30,54	21,35	51,89
	3	68	1221299	91,55	747,00	765	2,35	743,0	2,88	16,38	5,60	21,98
	4	113	1773390	93,44	826,89	855	3,29	814,3	4,76	23,84	8,11	31,95
	5	122	827631	89,52	684,56	701	2,35	692,5	1,21	23,13	3,78	26,91
12	2	55	867478	93,24	611,69	624	1,97	605,0	3,04	6,20	14,75	20,95
	3	26	206347	79,61	589,71	600	1,71	596,0	0,67	2,26	4,35	6,61
	4	25	189843	86,42	605,00	610	0,82	608,0	0,33	3,19	2,30	5,49
	5	33	180315	78,77	591,38	600	1,44	596,0	0,67	4,67	2,94	7,61
13	2	160	3587344	93,02	2707,00	2770	2,27	2714,0	2,02	29,69	19,20	48,89
	3	121	874665	84,43	2639,00	2639	0,00	2570,0	2,61	19,78	5,37	25,15
	4	114	462229	64,03	2702,37	2726	0,87	2673,0	1,94	30,08	2,97	33,05
	5	96	350325	80,75	2694,33	2735	1,49	2730,9	0,15	17,04	1,10	18,14
14	2	259	15134074	94,04	1229,89	1258	2,23	1242,5	1,23	134,81	265,09	399,90
	3	247	30366323	95,07	1191,05	1213	1,81	1194,2	1,55	276,95	1525,60	1802,55
	4	279	12750641	90,07	1178,03	1195	1,42	1166,0	2,43	266,49	446,60	713,09
	5	270	8497179	89,53	1297,70	1315	1,32	1289,1	1,97	283,09	139,62	422,71
15	2	353	73089549	96,72	1093,55	1134	3,57	1101,9	2,83	426,88	3284,32	3711,20
	3	247	44909743	96,12	1164,20	1198	2,82	1178,7	1,61	372,25	3319,05	3691,30
	4	154	3826316	90,60	1349,42	1379	2,15	1358,0	1,52	87,99	45,70	133,69
	5	180	6545862	90,78	1348,11	1370	1,60	1361,0	0,66	186,90	185,56	372,46
16	2	39	61057	77,76	681,25	689	1,12	682,0	1,02	0,84	0,41	1,25

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	3	63	185208	85,26	690,23	693	0,40	682,0	1,59	2,08	1,64	3,72
	4	69	217835	89,22	699,00	714	2,10	691,0	3,22	2,74	1,93	4,67
	5	34	75121	77,27	679,05	687	1,16	682,0	0,73	0,87	0,62	1,49
17	2	43	308126	90,15	850,13	851	0,10	851,0	<b>0,00</b>	1,80	4,02	5,82
	3	55	300295	89,66	841,83	842	0,02	842,0	<b>0,00</b>	1,14	4,38	5,52
	4	53	270124	88,54	843,33	845	0,20	844,0	0,12	2,03	3,34	5,37
	5	51	302996	89,96	841,83	842	0,02	842,0	<b>0,00</b>	1,66	4,42	6,08
18	2	372	15469412	91,36	1044,71	1083	3,54	-	-	218,76	172,61	391,37
	3	292	18732895	90,65	1078,83	1127	4,27	-	-	297,92	436,35	734,27
	4	263	43471567	95,52	1147,82	1169	1,81	-	-	475,59	1692,86	2168,45
	5	307	46785872	94,00	965,32	988	2,30	-	-	983,00	2960,06	3943,06
19	2	252	7966635	91,63	776,55	807	3,77	-	-	102,79	50,03	152,82
	3	322	18312122	94,87	796,34	818	2,65	-	-	174,98	177,20	352,18
	4	175	7024589	93,60	819,87	834	1,69	-	-	98,21	50,16	148,37
	5	164	2492648	91,46	735,41	745	1,29	-	-	56,91	7,27	64,18
20	2	396	65733339	93,30	547,23	597	8,34	-	-	769,67	3033,22	3802,89
	3	307	47160887	94,85	562,22	668	15,84	-	-	582,34	3275,39	3857,73
	4	424	77647116	96,78	560,70	588	4,64	-	-	641,84	3141,19	3783,03
	5	469	94167807	97,24	513,72	533	3,62	-	-	952,71	2686,99	3639,70
<b>Média</b>	-	-	-	83,24	-	-	2,04	-	2,10	-	-	-

**Tabela A.3:** Resultados detalhados do **CGV** para as Classes 2-5 com as *Características Ti* no caso irrestrito.

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	29	470636	99,31	288,33	316	8,76	273	13,61	0,57	6,23	6,80
	3	18	55623	94,79	278,50	280	0,54	279	0,36	0,41	0,30	0,71
	4	7	5598	77,49	290,00	290	0,00	277	4,48	0,15	0,01	0,16
	5	5	1707	68,83	277,00	279	0,72	-	-	0,03	0,07	0,10
2	2	4	2738	82,43	321,33	336	4,37	329	2,08	0,02	0,10	0,12
	3	3	3045	83,88	336,25	347	3,10	347	<b>0,00</b>	0,02	0,14	0,16

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	1	1428	68,21	321,33	336	4,37	329	2,08	0,02	0,11	0,13
	5	4	4446	87,45	321,33	329	2,33	329	<b>0,00</b>	0,01	0,13	0,14
3	2	18	29435	86,24	386,83	393	1,57	381	3,05	0,35	0,17	0,52
	3	9	15988	83,89	399,00	399	0,00	387	3,01	0,32	0,03	0,35
	4	22	17388	87,45	368,75	374	1,40	-	-	0,11	0,19	0,30
	5	16	12724	85,39	364,50	369	1,22	369	<b>0,00</b>	0,08	2,16	2,24
4	2	16	20765	88,90	432,00	432	0,00	423	2,08	0,16	0,10	0,26
	3	15	24048	88,95	433,50	438	1,03	423	3,42	0,18	0,29	0,47
	4	15	50939	95,28	438,00	438	0,00	438	<b>0,00</b>	0,31	0,34	0,65
	5	17	29075	90,92	423,00	423	0,00	423	<b>0,00</b>	0,27	0,14	0,41
5	2	76	120105	93,15	371,00	371	0,00	367	1,08	0,82	0,24	1,06
	3	53	426305	95,43	382,00	385	0,78	373	3,12	2,60	5,88	8,48
	4	72	213027	93,22	376,14	377	0,23	377	<b>0,00</b>	1,47	1,61	3,08
	5	60	61839	92,84	382,00	389	1,80	389	<b>0,00</b>	0,22	0,25	0,47
6	2	35	53131	91,63	482,80	489	1,27	488	0,20	0,32	0,38	0,70
	3	24	65280	93,21	491,50	500	1,70	489	2,20	0,44	0,49	0,93
	4	30	58410	95,08	494,80	497	0,44	489	1,61	0,38	0,33	0,71
	5	33	86904	96,07	472,50	488	3,18	488	<b>0,00</b>	0,18	0,55	0,73
7	2	59	232296	94,34	728,33	744	2,11	-	-	1,57	0,72	2,29
	3	41	53064	80,46	695,00	695	0,00	685	1,44	1,52	0,26	1,78
	4	36	133853	94,67	713,00	722	1,25	-	-	1,12	0,40	1,52
	5	23	21128	82,24	737,00	742	0,67	742	<b>0,00</b>	0,43	0,12	0,55
8	2	84	260048	94,80	696,00	708	1,69	680	3,95	1,47	0,81	2,28
	3	52	193320	94,70	747,67	762	1,88	-	-	1,52	0,76	2,28
	4	50	90949	86,41	713,00	722	1,25	-	-	2,03	0,46	2,49
	5	15	12444	87,70	721,00	721	0,00	-	-	0,07	0,06	0,13
9	2	19	70587	91,87	616,50	622	0,88	609	2,09	0,62	2,26	2,88
	3	28	151257	95,82	623,13	626	0,46	609	2,72	0,96	1,59	2,55
	4	30	79385	92,88	616,75	635	2,87	626	1,42	0,51	1,08	1,59
	5	29	70654	93,18	609,00	609	0,00	609	<b>0,00</b>	0,36	0,57	0,93
10	2	207	7105950	98,88	682,86	726	5,94	-	-	13,83	48,59	62,42

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	3	175	10026109	98,94	633,83	677	6,38	-	-	19,69	127,40	147,09
	4	111	2975361	98,50	745,46	787	5,28	-	-	7,90	24,54	32,44
	5	128	688886	95,60	694,73	710	2,15	-	-	5,74	2,42	8,16
11	2	149	2474691	98,08	675,43	728	7,22	-	-	6,87	18,03	24,90
	3	117	1994486	98,34	722,50	734	1,57	-	-	5,56	9,87	15,43
	4	85	2079692	98,42	813,38	834	2,47	-	-	6,59	16,70	23,29
	5	109	650481	95,27	677,86	696	2,61	-	-	6,08	2,86	8,94
12	2	58	491898	98,16	607,56	619	1,85	596	3,72	0,95	6,10	7,05
	3	36	335033	96,85	589,71	597	1,22	596	0,17	0,44	7,88	8,32
	4	46	221207	96,70	598,50	602	0,58	-	-	0,72	2,25	2,97
	5	23	219671	95,36	591,00	597	1,01	-	-	0,89	4,92	5,81
13	2	107	1155853	93,53	2707,34	2768	2,19	-	-	8,49	8,83	17,32
	3	121	2112562	95,12	2571,00	2602	1,19	-	-	17,13	15,61	32,74
	4	135	1696385	96,19	2665,00	2717	1,91	-	-	11,79	10,96	22,75
	5	79	450395	93,04	2656,25	2679	0,85	-	-	6,35	1,28	7,63
14	2	262	18365939	97,67	1203,04	1230	2,19	-	-	76,46	401,92	478,38
	3	247	21438238	97,89	1183,88	1206	1,83	-	-	97,10	1354,22	1451,32
	4	296	13210751	97,15	1171,02	1177	0,51	-	-	81,75	460,33	542,08
	5	195	3099258	95,03	1287,50	1288	0,04	-	-	37,41	25,01	62,42
15	2	217	39915544	98,34	1074,98	1105	2,72	-	-	134,45	3527,89	3662,34
	3	159	37460135	98,63	1155,48	1191	2,98	-	-	136,01	3497,35	3633,36
	4	231	14936903	98,58	1338,55	1366	2,01	-	-	60,99	399,34	460,33
	5	237	8667356	97,77	1334,24	1355	1,53	-	-	51,54	296,16	347,70
16	2	39	74676	89,50	679,33	682	0,39	682	<b>0,00</b>	0,44	0,53	0,97
	3	50	105723	92,00	679,25	682	0,40	682	<b>0,00</b>	0,47	0,69	1,16
	4	33	84042	91,17	691,00	693	0,29	687	0,87	0,61	0,60	1,21
	5	34	60698	85,76	679,05	682	0,43	682	<b>0,00</b>	0,26	0,64	0,90
17	2	34	255248	95,32	849,57	851	0,17	-	-	0,61	4,49	5,10
	3	51	379355	96,87	841,83	842	0,02	-	-	0,43	5,95	6,38
	4	49	299496	96,37	842,00	842	0,00	-	-	0,54	3,94	4,48
	5	51	327635	96,47	841,83	842	0,02	-	-	0,55	4,64	5,19

I	Cl	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
18	2	268	22180899	97,89	1021,43	1061	3,73	-	-	85,44	466,66	552,10
	3	308	32549738	97,74	1060,85	1109	4,34	-	-	156,81	1485,90	1642,71
	4	239	28514916	98,26	1124,21	1158	2,92	-	-	123,38	661,27	784,65
	5	274	51515734	98,35	950,14	985	3,54	-	-	285,10	3344,29	3629,39
19	2	290	13839442	96,66	773,00	806	4,09	-	-	80,11	129,03	209,14
	3	167	6432279	95,43	780,74	806	3,13	-	-	55,55	55,48	111,03
	4	166	4620911	95,49	805,56	833	3,29	-	-	44,21	33,37	77,58
	5	196	2153850	93,70	724,56	734	1,29	-	-	33,95	7,73	41,68
20	2	279	58446518	97,86	534,38	581	8,02	-	-	260,73	3638,22	3898,95
	3	356	63435363	98,30	524,15	549	4,53	-	-	269,61	3508,62	3778,23
	4	466	100018989	99,01	547,00	561	2,50	-	-	302,10	3311,32	3613,42
	5	518	89662437	98,75	503,18	512	1,72	-	-	367,22	3289,06	3656,28
<b>Média</b>	-	-	-	93,10	-	-	1,94	-	1,68	-	-	-

**Tabela A.4:** Resultados detalhados do CGN para as Classes 2-5 com as *Características Ti* no caso sequencial.

I	Cl	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	55	2	267,60	285	6,11	285,0	<b>0,00</b>	0,04	5,09	5,13
	3	20	89	21	272,50	315	13,49	280,0	11,11	5,43	26,28	31,71
	4	3	31	2	284,67	290	1,84	288,0	0,69	0,29	0,20	0,49
	5	1	6	0	279,00	279	0,00	279,0	<b>0,00</b>	0,00	0,03	0,03
2	2	9	29	8	331,33	347	4,51	342,0	1,44	0,01	1,23	1,24
	3	4	21	3	335,00	350	4,29	347,0	0,86	0,03	0,48	0,51
	4	1	6	0	321,33	336	4,37	336,0	<b>0,00</b>	0,00	0,11	0,11
	5	1	3	0	321,33	329	2,33	329,0	<b>0,00</b>	0,00	0,07	0,07
3	2	44	145	60	377,25	405	6,85	389,0	3,95	0,06	12,36	12,42
	3	4	83	7	381,00	393	3,05	387,0	1,53	1,70	0,67	2,37
	4	4	39	3	366,50	374	2,01	374,0	<b>0,00</b>	0,65	0,38	1,03
	5	1	31	0	364,50	369	1,22	369,0	<b>0,00</b>	0,00	0,14	0,14

I	Cl	$\#it_{EL}$	$\#it_{CG}$	$\#Cols_{rej}$	RL	LS	$gap_{RL}(\%)$	$lit^*$	$gap_{lit^*}(\%)$	$T_{car}(s)$	$T_{rot}(s)$	$T_{tot}(s)$
4	2	4	31	3	428,67	434	1,23	434,0	<b>0,00</b>	0,01	0,68	0,69
	3	2	29	1	428,67	434	1,23	432,0	0,46	0,46	0,43	0,89
	4	1	16	0	437,56	438	0,10	438,0	<b>0,00</b>	0,00	0,43	0,43
	5	1	21	0	423,00	423	0,00	423,0	<b>0,00</b>	0,00	0,05	0,05
5	2	6	105	9	375,00	382	1,83	380,0	0,52	0,21	2,87	3,08
	3	3	90	2	373,00	373	0,00	373,0	<b>0,00</b>	0,02	1,23	1,25
	4	3	78	2	375,00	377	0,53	377,0	<b>0,00</b>	1,12	1,27	2,39
	5	1	56	0	382,00	389	1,80	389,0	<b>0,00</b>	0,00	0,14	0,14
6	2	1	33	0	472,50	491	3,77	491,0	<b>0,00</b>	0,00	0,32	0,32
	3	3	69	2	474,00	496	4,44	496,0	<b>0,00</b>	0,30	0,96	1,26
	4	1	48	0	481,67	495	2,69	489,0	1,21	0,10	0,77	0,87
	5	1	19	0	472,50	492	3,96	488,0	0,81	5,97	0,55	6,52
7	2	7	108	7	701,50	748	6,22	724,0	3,21	0,10	2,33	2,43
	3	5	134	6	687,50	698	1,50	698,0	<b>0,00</b>	0,65	1,03	1,68
	4	5	73	4	707,44	723	2,15	714,0	1,24	3,19	1,28	4,47
	5	1	35	0	735,00	743	1,08	742,0	0,13	0,68	0,26	0,94
8	2	11	150	14	690,20	720	4,14	720,0	<b>0,00</b>	0,03	6,06	6,09
	3	9	125	14	721,67	738	2,21	730,0	1,08	1,98	1,97	3,95
	4	2	100	1	695,50	717	3,00	701,0	2,23	126,44	0,55	126,99
	5	1	49	0	721,00	721	0,00	721,0	<b>0,00</b>	2,74	0,08	2,82
9	2	2	36	1	612,00	612	0,00	612,0	<b>0,00</b>	0,00	0,57	0,57
	3	3	34	2	615,00	615	0,00	615,0	<b>0,00</b>	1,59	0,64	2,23
	4	4	34	3	616,00	633	2,69	626,0	1,11	0,24	2,23	2,47
	5	1	15	0	609,00	609	0,00	609,0	<b>0,00</b>	0,00	0,21	0,21
10	2	14	274	25	633,93	717	11,59	687,0	4,18	0,17	153,19	153,36
	3	23	284	39	610,50	685	10,88	638,0	6,86	12,79	65,42	78,21
	4	10	211	14	706,94	776	8,90	738,0	4,90	10,47	50,96	61,43
	5	4	132	4	683,10	724	5,65	706,0	2,49	31,89	3,55	35,44
11	2	16	255	39	629,00	753	16,47	708,8	5,87	0,14	94,43	94,57
	3	8	197	17	695,17	765	9,13	743,0	2,88	1,10	4,50	5,60
	4	81	412	174	754,96	885	14,69	814,3	7,99	40,65	381,96	422,61

I	Cl	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	5	5	171	8	669,10	691	3,17	692,5	<b>-0,22</b>	88,78	1,81	90,59
12	2	8	81	13	597,50	605	1,24	605,0	<b>0,00</b>	0,03	36,42	36,45
	3	1	28	0	589,25	600	1,79	596,0	0,67	0,00	5,93	5,93
	4	5	54	5	595,00	608	2,14	608,0	<b>0,00</b>	2,66	8,71	11,37
	5	1	41	0	589,25	600	1,79	596,0	0,67	0,00	7,51	7,51
13	2	10	248	26	2547,08	2761	7,75	2714,0	1,70	0,11	20,34	20,45
	3	26	317	57	2534,17	2606	2,76	2570,0	1,38	5,49	106,38	111,87
	4	10	185	17	2560,29	2679	4,43	2673,0	0,22	4,35	36,17	40,52
	5	12	186	19	2655,82	2735	2,90	2730,9	0,15	567,57	9,90	577,47
14	2	24	424	53	1182,09	1259	6,11	1242,5	1,31	0,43	1208,75	1209,18
	3	14	430	18	1155,83	1224	5,57	1194,2	2,43	4,38	1522,37	1526,75
	4	17	445	22	1158,97	1215	4,61	1166,0	4,03	18,26	1640,98	1659,24
	5	2	295	1	1287,50	1288	0,04	1289,1	<b>-0,09</b>	414,01	27,95	441,96
15	2	1	380	3	1047,54	-	-	1101,9	-	0,00	14400,68	14400,68
	3	36	446	86	1093,80	-	-	1178,7	-	15,35	14385,32	14400,67
	4	3	290	5	1320,57	1356	2,61	1358,0	<b>-0,15</b>	4,98	62,27	67,25
	5	2	285	2	1320,94	1344	1,72	1361,0	<b>-1,26</b>	125,14	91,07	216,21
16	2	2	48	1	679,05	688	1,30	682,0	0,87	0,01	1,13	1,14
	3	1	52	0	679,05	689	1,44	682,0	1,02	0,00	0,77	0,77
	4	5	85	7	686,20	695	1,27	691,0	0,58	5,23	1,72	6,95
	5	1	34	0	679,05	687	1,16	682,0	0,73	0,00	0,64	0,64
17	2	2	64	3	849,64	851	0,16	851,0	<b>0,00</b>	0,01	8,60	8,61
	3	1	55	0	841,83	842	0,02	842,0	<b>0,00</b>	0,00	4,46	4,46
	4	1	54	0	841,83	842	0,02	844,0	<b>-0,24</b>	0,12	4,05	4,17
	5	1	51	0	841,83	842	0,02	842,0	<b>0,00</b>	0,00	4,39	4,39
18	2	8	391	34	951,59	-	-	-	-	0,83	14399,81	14400,64
	3	8	457	49	976,17	-	-	-	-	63,56	14336,81	14400,37
	4	12	459	43	1043,41	-	-	-	-	30,59	14370,38	14400,97
	5	3	395	4	919,49	-	-	-	-	31,28	14369,16	14400,44
19	2	80	699	340	728,55	830	12,22	-	-	4,35	996,28	1000,63
	3	136	972	636	762,75	899	15,16	-	-	7,16	3544,93	3552,09



I	Cl	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	82	578	261	779,19	845	7,79	-	-	126,51	551,51	678,02
	5	8	284	19	720,88	739	2,45	-	-	749,44	16,22	765,66
20	2	1	820	9	474,33	-	-	-	-	0,00	14400,62	14400,62
	3	1	679	9	487,98	-	-	-	-	0,01	14400,65	14400,66
	4	3	646	23	520,55	-	-	-	-	8,25	14393,00	14401,25
	5	6	751	20	489,83	-	-	-	-	696,87	13713,19	14410,06
21	2	1	1003	12	925,46	-	-	-	-	0,00	3603,19	3603,19
	3	8	757	84	1067,70	-	-	-	-	16,75	3585,08	3601,83
	4	10	909	86	989,93	-	-	-	-	2,97	3612,93	3615,90
	5	14	791	34	957,89	1005	4,69	-	-	349,54	2410,12	2759,66
22	2	1	863	11	939,61	-	-	-	-	0,01	3600,55	3600,56
	3	1	732	10	994,27	-	-	-	-	0,01	3601,89	3601,90
	4	1	657	11	1021,68	-	-	-	-	0,00	3600,94	3600,94
	5	11	768	31	981,77	1042	5,78	-	-	1358,86	1969,12	3327,98
<b>Média</b>	-	-	-	29,10	-	-	3,81	-	1,22	-	-	-

**Tabela A.5:** Resultados detalhados do CGN para as Classes 2-5 com as Características  $T_i$  no caso irrestrito.

I	Cl	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	1	29	0	267,60	289	7,40	273	5,54	0,03	2,31	2,34
	3	3	46	2	270,00	280	3,57	279	0,36	23,58	4,10	27,68
	4	1	11	0	277,00	277	0,00	277	<b>0,00</b>	0,01	0,04	0,05
	5	2	7	1	277,00	279	0,72	-	-	18,91	0,13	19,04
2	2	1	5	0	321,33	336	4,37	329	2,08	0,00	0,09	0,09
	3	2	11	1	334,86	347	3,50	347	<b>0,00</b>	0,15	0,72	0,87
	4	1	1	0	321,33	336	4,37	329	2,08	0,00	0,08	0,08
	5	1	4	0	321,33	329	2,33	329	<b>0,00</b>	0,00	0,10	0,10
3	2	5	46	7	371,00	386	3,89	381	1,30	0,01	0,76	0,77
	3	2	45	2	380,91	387	1,57	387	<b>0,00</b>	7,84	0,20	8,04
	4	1	28	0	365,75	374	2,21	-	-	0,00	0,57	0,57

<b>I</b>	<b>Cl</b>	$\#it_{EL}$	$\#it_{CG}$	$\#Cols_{rej}$	RL	LS	$gap_{RL}(\%)$	$lit^*$	$gap_{lit^*}(\%)$	$T_{car}(s)$	$T_{rot}(s)$	$T_{tot}(s)$
	5	1	17	0	364,50	369	1,22	369	<b>0,00</b>	0,00	0,26	0,26
4	2	1	22	0	423,00	423	0,00	423	<b>0,00</b>	0,00	0,08	0,08
	3	1	19	0	423,00	423	0,00	423	<b>0,00</b>	0,39	0,18	0,57
	4	1	17	0	437,56	438	0,10	438	<b>0,00</b>	0,00	0,55	0,55
	5	1	17	0	423,00	423	0,00	423	<b>0,00</b>	0,00	0,19	0,19
5	2	1	49	0	366,25	367	0,20	367	<b>0,00</b>	0,02	0,24	0,26
	3	3	107	2	373,00	373	0,00	373	<b>0,00</b>	0,19	1,45	1,64
	4	3	74	2	375,00	380	1,32	377	0,79	10,31	1,73	12,04
	5	1	67	0	382,00	389	1,80	389	<b>0,00</b>	0,00	0,23	0,23
6	2	1	54	0	472,50	489	3,37	488	0,20	0,00	0,69	0,69
	3	2	50	1	474,00	492	3,66	489	0,61	0,13	0,74	0,87
	4	1	51	0	481,67	490	1,70	489	0,20	0,05	0,88	0,93
	5	1	33	0	472,50	488	3,18	488	<b>0,00</b>	0,00	0,53	0,53
7	2	4	95	3	697,83	715	2,40	-	-	0,01	0,91	0,92
	3	1	55	0	682,50	685	0,36	685	<b>0,00</b>	1,68	0,19	1,87
	4	2	57	1	706,67	722	2,12	-	-	8,04	0,51	8,55
	5	1	38	0	735,00	742	0,94	742	<b>0,00</b>	0,01	0,14	0,15
8	2	1	65	0	680,00	680	0,00	680	<b>0,00</b>	0,00	0,16	0,16
	3	2	66	1	719,67	729	1,28	-	-	1,36	0,55	1,91
	4	1	67	0	695,50	711	2,18	-	-	2,18	0,20	2,38
	5	1	15	0	721,00	721	0,00	-	-	0,00	0,04	0,04
9	2	1	20	0	609,00	609	0,00	609	<b>0,00</b>	0,00	0,11	0,11
	3	1	37	0	609,00	609	0,00	609	<b>0,00</b>	0,00	0,27	0,27
	4	2	53	1	616,00	626	1,60	626	<b>0,00</b>	1,01	2,18	3,19
	5	1	31	0	609,00	609	0,00	609	<b>0,00</b>	0,00	0,62	0,62
10	2	1	294	0	631,29	681	7,30	-	-	0,00	58,75	58,75
	3	7	207	12	608,90	675	9,79	-	-	24,09	16,40	40,49
	4	2	217	1	706,85	737	4,09	-	-	8,84	21,06	29,90
	5	1	117	0	681,50	713	4,42	-	-	1,69	0,91	2,60
11	2	6	269	12	629,56	728	13,52	-	-	0,65	43,35	44,00
	3	3	205	4	687,79	733	6,17	-	-	18,84	3,77	22,61

I	Cl	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	10	227	14	742,77	778	4,53	-	-	73,46	46,21	119,67
	5	1	136	0	668,50	683	2,12	-	-	0,34	0,63	0,97
12	2	7	76	7	591,30	610	3,07	596	2,30	0,05	27,54	27,59
	3	1	38	0	589,25	596	1,13	596	<b>0,00</b>	0,00	8,49	8,49
	4	3	68	2	593,00	604	1,82	-	-	9,78	4,68	14,46
	5	1	31	0	589,25	596	1,13	-	-	117,33	8,12	125,45
13	2	4	187	4	2542,69	2707	6,07	-	-	0,13	7,47	7,60
	3	6	191	10	2513,50	2548	1,35	-	-	99,44	29,39	128,83
	4	4	176	4	2559,50	2686	4,71	-	-	55,38	11,06	66,44
	5	2	119	2	2632,00	2632	0,00	-	-	637,41	1,07	638,48
14	2	4	399	4	1178,41	1230	4,19	-	-	2,42	264,22	266,64
	3	2	372	3	1158,88	1206	3,91	-	-	3,34	314,73	318,07
	4	3	423	2	1155,00	1173	1,53	-	-	90,56	401,41	491,97
	5	1	235	0	1287,50	1288	0,04	-	-	0,00	27,37	27,37
15	2	1	320	3	1047,36	-	-	-	-	0,01	14400,42	14400,43
	3	7	366	16	1090,66	1191	8,43	-	-	125,84	5700,32	5826,16
	4	1	230	0	1320,50	1336	1,16	-	-	1591,98	49,26	1641,24
	5	1	230	0	1320,53	1336	1,16	-	-	140,09	81,25	221,34
16	2	1	44	0	679,05	682	0,43	682	<b>0,00</b>	0,01	0,83	0,84
	3	1	51	0	679,05	682	0,43	682	<b>0,00</b>	0,00	1,03	1,03
	4	3	39	2	679,60	689	1,36	687	0,29	2,71	1,20	3,91
	5	1	34	0	679,05	682	0,43	682	<b>0,00</b>	0,00	0,54	0,54
17	2	2	44	1	848,62	851	0,28	-	-	0,00	6,31	6,31
	3	1	51	0	841,83	842	0,02	-	-	0,00	5,90	5,90
	4	1	59	0	841,83	842	0,02	-	-	0,00	4,75	4,75
	5	1	51	0	841,83	842	0,02	-	-	0,00	4,70	4,70
18	2	5	354	14	952,22	1047	9,05	-	-	11,73	11141,87	11153,60
	3	5	436	23	992,80	-	-	-	-	127,26	14273,12	14400,38
	4	10	417	25	1047,93	1157	9,43	-	-	729,73	10881,04	11610,77
	5	2	294	2	919,46	978	5,99	-	-	622,49	8169,82	8792,31
19	2	45	545	164	722,77	824	12,29	-	-	50,27	615,06	665,33

<b>I</b>	<b>Cl</b>	$\#it_{EL}$	$\#it_{CG}$	$\#Cols_{rej}$	RL	LS	$gap_{RL}(\%)$	$lit^*$	$gap_{lit^*}(\%)$	$T_{car}(s)$	$T_{rot}(s)$	$T_{tot}(s)$
	3	57	485	179	746,43	823	9,30	-	-	71,53	954,84	1026,37
	4	11	353	21	764,70	786	2,71	-	-	427,80	46,46	474,26
	5	4	229	3	718,56	728	1,30	-	-	3061,86	13,85	3075,71
21	2	1	927	12	926,69	-	-	-	-	0,00	3600,66	3600,66
	3	6	652	40	1064,50	-	-	-	-	144,57	3458,23	3602,80
	4	12	783	76	990,37	-	-	-	-	74,99	3533,71	3608,70
	5	2	737	1	956,67	982	2,58	-	-	1628,79	501,24	2130,03
24	2	4	716	45	1090,84	-	-	-	-	0,39	3604,06	3604,45
	3	1	590	10	1045,69	-	-	-	-	0,01	3600,77	3600,78
	4	1	558	8	1062,16	-	-	-	-	0,00	3601,25	3601,25
	5	2	651	2	1030,19	1046	1,51	-	-	283,20	2097,82	2381,02
<b>Média</b>	-	-	-	8,99	-	-	2,77	-	0,45	-	-	-

## A.2 Resultados para as Instâncias com *Características Tr*

As Tabelas A.6-A.10 apresentam os resultados dos algoritmos *CGV* e *CGN* para as instâncias com *Características Tr*. Suas colunas têm o mesmo significado daquelas apresentadas na Seção anterior. Além dessas colunas em comum, a coluna *#VU* é utilizada para apresentar o número de veículos usados nas soluções obtidas.

**Tabela A.6:** Resultados detalhados do *CGV/CGN* para a Classe 1 com as *Características Tr*.

I	#VU	<i>#it<sub>CG</sub></i>	RL	LS	<i>gap<sub>RL</sub></i> (%)	<i>lit*</i>	<i>gap<sub>lit*</sub></i> (%)	<i>T<sub>tot</sub></i> (s)
1	3	20	271,13	278,73	2,73	278,73	<b>0,00</b>	2,03
2	5	6	327,44	334,96	2,25	334,96	<b>0,00</b>	0,10
3	4	32	351,82	370,80	5,12	358,40	3,34	0,80
4	6	13	430,89	430,89	0,00	430,88	<b>0,00</b>	0,07
5	4	25	374,13	375,28	0,31	375,28	<b>0,00</b>	0,54
6	6	29	482,01	495,85	2,79	495,85	<b>0,00</b>	0,50
7	3	257	559,81	568,56	1,54	568,56	<b>0,00</b>	128,50
8	3	502	559,81	568,56	1,54	568,56	<b>0,00</b>	154,41
9	8	26	607,65	607,65	0,00	607,65	<b>0,00</b>	2,65
10	3	275	486,19	539,69	9,91	535,80	0,72	1246,64
11	4	383	486,19	511,64	4,97	505,01	1,30	1081,30
12	9	38	603,23	614,94	1,91	610,00	0,80	14,66
13	3	606	2006,68	2006,68	0,00	2006,34	0,02	3600,08
14	4	229	833,02	859,19	3,05	837,67	2,51	3600,36
15	4	347	830,80	855,00	2,83	837,67	2,03	3600,32
16	11	37	695,33	698,61	0,47	698,61	<b>0,00</b>	0,80
17	14	47	861,09	861,79	0,08	861,79	<b>0,00</b>	5,07
18	4	805	743,74	743,74	0,00	723,54	2,72	3600,04
19	5	353	542,07	549,60	1,37	524,61	4,55	3600,78
20	4	2264	257,75	257,99	0,09	241,97	6,21	3600,10
21	7	1549	692,79	701,96	1,31	687,6	2,05	3600,35
22	8	1378	751,60	770,58	2,46	740,66	3,88	3600,88
23	10	565	865,31	878,49	1,50	835,26	4,92	3600,52
24	14	245	1040,18	1060,18	1,89	1026,6	3,17	3600,33
25	8	2274	843,41	849,78	0,75	826,14	2,78	3600,43
26	10	2330	819,87	820,58	0,09	819,56	0,12	3600,30
27	14	694	1108,04	1141,16	2,90	1082,65	5,13	3600,96
28	7	3648	1086,02	1233,17	11,93	1040,7	15,61	3604,46
29	7	4799	1240,24	1274,19	2,66	1162,23	8,79	3603,79
30	12	2683	1078,84	1082,66	0,35	1033,42	4,55	3601,27
31	16	576	1644,51	1690,52	2,72	1306,07	22,74	3600,87
32	17	4382	1359,47	1418,05	4,13	1303,52	8,08	3836,11
33	17	4665	1368,44	1382,19	1,00	1301,06	5,87	3603,80
34	22	2247	749,90	807,30	7,11	711,98	11,81	3701,24
35	27	2755	917,54	997,58	8,02	868,12	12,98	4372,69
36	14	10900	647,74	670,31	3,37	586,58	12,49	3613,84
<b>Média</b>	-	-	-	-	2,59	-	4,14	-

Tabela A.7: Resultados detalhados do CGV para as Classes 2-5 com as Características Tr no caso sequencial.

I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	28	548032	87,50	303,56	320,00	5,14	291,49	8,91	8,78	9,20	17,98
	3	3	25	371512	86,73	305,00	315,99	3,48	306,03	3,15	7,55	4,79	12,34
	4	4	12	13860	75,67	296,75	296,75	0,00	296,75	0,00	0,43	0,05	0,48
	5	4	11	13343	69,58	279,09	285,93	2,40	284,23	0,60	0,77	0,21	0,98
2	2	5	2	3154	58,47	342,69	354,01	3,20	347,73	1,77	0,09	0,10	0,19
	3	5	1	1985	43,32	347,45	356,24	2,47	356,24	0,00	0,10	0,09	0,19
	4	5	8	6006	72,63	327,60	342,00	4,21	342,00	<b>0,00</b>	0,15	0,12	0,27
	5	5	3	2124	59,60	327,44	334,96	2,25	334,96	<b>0,00</b>	0,04	0,09	0,13
3	2	5	18	32537	64,77	406,87	414,39	1,82	403,93	2,52	1,15	0,26	1,41
	3	5	25	138628	85,59	417,97	417,97	0,00	394,72	5,56	3,37	0,50	3,87
	4	5	35	54877	77,95	373,27	375,60	0,62	372,56	0,81	1,27	0,26	1,53
	5	5	48	102314	83,65	354,98	365,34	2,83	358,40	1,90	1,88	0,53	2,41
4	2	6	11	27114	68,50	451,98	451,98	0,00	440,94	2,44	0,74	0,10	0,84
	3	6	18	37650	74,07	442,20	446,61	0,99	445,25	0,30	1,04	0,20	1,24
	4	6	10	30962	68,64	447,37	447,37	0,00	447,37	<b>0,00</b>	1,45	0,17	1,62
	5	6	17	15026	69,63	430,89	430,89	0,00	430,88	<b>0,00</b>	0,52	0,04	0,56
5	2	4	85	1405026	87,79	393,46	393,46	0,00	388,72	1,20	21,39	18,48	39,87
	3	4	101	4150446	91,43	398,77	398,77	0,00	383,15	3,92	57,65	127,48	185,13
	4	4	58	914033	87,63	392,00	408,16	3,96	385,70	5,50	15,73	13,17	28,90
	5	4	117	346659	88,91	375,28	375,28	0,00	375,28	<b>0,00</b>	4,69	1,62	6,31
6	2	6	30	129968	87,54	493,80	499,08	1,06	499,08	<b>0,00</b>	1,51	0,74	2,25
	3	6	37	165340	89,83	504,68	504,68	0,00	504,68	<b>0,00</b>	2,22	0,79	3,01
	4	6	31	76801	86,19	509,66	512,14	0,48	501,04	2,17	1,83	0,29	2,12
	5	6	18	71965	87,08	482,01	502,75	4,12	495,85	1,37	0,86	0,47	1,33
7	2	5	47	365701	86,74	749,67	767,39	2,31	734,65	4,27	6,19	1,44	7,63
	3	5	58	334377	82,25	709,72	709,72	0,00	702,59	1,00	9,19	2,01	11,20
	4	5	44	152477	70,76	722,49	722,49	0,00	703,64	2,61	8,50	0,65	9,15
	5	5	67	377871	76,80	681,72	702,26	2,93	669,17	4,71	22,62	2,36	24,98
8	2	5	45	272277	77,57	720,15	731,32	1,53	727,40	0,54	7,11	1,37	8,48
	3	5	33	200495	81,01	770,84	772,68	0,24	741,12	4,08	5,90	0,97	6,87

I	CI	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	5	55	470234	86,86	718,81	736,73	2,43	723,55	1,79	11,76	3,13	14,89
	5	4	144	1740548	88,34	630,20	656,54	4,01	646,46	1,54	51,36	23,54	74,90
9	2	8	21	28598	76,87	611,49	611,49	0,00	611,49	0,00	0,60	0,08	0,68
	3	8	19	68083	67,24	640,63	640,63	0,00	622,15	2,89	3,74	0,51	4,25
	4	8	30	116444	75,94	615,24	632,14	2,67	625,13	1,11	3,55	1,51	5,06
	5	8	15	33427	76,70	607,65	607,65	0,00	607,65	<b>0,00</b>	0,79	0,17	0,96
10	2	6	136	7764902	94,44	720,24	788,26	8,63	700,20	11,17	61,20	100,70	161,90
	3	5	237	18503514	94,90	645,64	673,39	4,12	636,80	5,43	169,50	378,66	548,16
	4	7	176	6163762	95,17	761,39	806,55	5,60	763,98	5,28	60,11	90,84	150,95
	5	6	192	11455867	95,00	662,43	719,04	7,87	705,62	1,87	169,67	193,27	362,94
11	2	6	180	3438061	92,34	710,35	763,87	7,01	722,44	5,42	33,77	29,31	63,08
	3	7	87	2640348	93,87	754,73	779,73	3,21	730,30	6,34	26,73	22,92	49,65
	4	7	115	2048565	94,03	841,38	870,87	3,39	832,91	4,36	25,16	9,74	34,90
	5	6	195	21808939	96,83	663,81	666,80	0,45	649,75	2,56	221,24	662,60	883,84
12	2	9	61	993968	93,97	627,48	638,06	1,66	620,47	2,76	5,74	15,28	21,02
	3	9	30	133034	71,87	603,98	615,41	1,86	610,00	0,88	2,12	2,43	4,55
	4	9	23	101997	81,29	617,74	623,21	0,88	623,71	<b>-0,08</b>	2,25	0,85	3,10
	5	9	34	201351	77,33	606,35	614,94	1,40	610,23	0,77	6,19	3,74	9,93
13	2	7	165	8097567	94,40	2706,80	2783,82	2,77	2669,39	4,11	56,57	108,33	164,90
	3	6	207	16614632	94,18	2609,92	2629,87	0,76	2500,55	4,92	180,42	515,13	695,55
	4	7	160	3070055	85,33	2696,10	2743,53	1,73	2703,48	1,46	97,83	50,11	147,94
	5	6	208	6314764	86,29	2472,32	2557,13	3,32	2490,87	2,59	290,05	279,87	569,92
14	2	6	266	36164293	94,92	1108,68	1192,28	7,01	1139,70	4,41	350,18	3250,26	3600,44
	3	6	179	31985228	94,44	1112,11	1157,35	3,91	1101,96	4,79	396,23	3204,28	3600,51
	4	6	156	34453120	95,54	1048,22	1155,41	9,28	1000,76	13,38	393,82	3206,78	3600,60
	5	5	211	33175510	95,49	1005,64	1055,76	4,75	959,51	9,12	641,84	2958,52	3600,36
15	2	6	162	44285305	96,70	1118,99	1159,35	3,48	1116,51	3,69	292,57	3307,92	3600,49
	3	6	226	39258099	95,01	1178,21	1259,48	6,45	1187,59	5,71	438,07	3162,40	3600,47
	4	7	205	42798152	96,48	1262,94	1316,93	4,10	1288,84	2,13	439,20	3161,14	3600,34
	5	7	195	35983621	96,92	1250,15	1305,83	4,26	1265,14	3,12	414,00	3186,33	3600,33
16	2	11	49	83695	83,42	697,56	704,80	1,03	698,60	0,88	0,87	0,45	1,32

I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	3	11	64	228838	88,16	706,23	708,83	0,37	698,61	1,44	2,10	2,01	4,11
	4	11	67	215749	87,71	716,46	731,13	2,01	709,27	2,99	3,12	1,94	5,06
	5	11	38	90714	79,98	695,33	703,35	1,14	698,60	0,68	0,79	0,89	1,68
17	2	14	51	380298	91,96	870,70	870,86	0,02	870,86	<b>0,00</b>	1,90	4,92	6,82
	3	14	60	264938	89,37	861,09	861,79	0,08	861,79	<b>0,00</b>	0,93	3,40	4,33
	4	14	49	243484	87,06	862,62	862,62	0,00	862,62	0,00	2,10	3,05	5,15
	5	14	56	400336	92,57	861,09	861,79	0,08	861,79	<b>0,00</b>	1,51	6,24	7,75
18	2	9	490	49700334	94,95	1058,19	1095,84	3,44	1059,71	3,30	440,46	1815,58	2256,04
	3	10	354	49770036	95,34	1097,39	1149,22	4,51	1122,90	2,29	461,16	3139,35	3600,51
	4	10	268	45474515	95,64	1166,38	1192,60	2,20	1163,80	2,41	492,15	2218,39	2710,54
	5	9	305	53877077	95,81	965,53	997,23	3,18	946,78	5,06	814,56	2785,74	3600,30
19	2	11	267	24389347	95,10	793,86	824,15	3,67	798,35	3,13	198,80	333,16	531,96
	3	11	336	27193666	94,51	815,18	839,21	2,86	802,99	4,32	283,85	450,65	734,50
	4	12	203	19545091	95,02	837,28	860,81	2,73	832,68	3,27	236,01	313,04	549,05
	5	9	413	70987978	96,67	686,35	708,22	3,09	683,12	3,54	921,78	2678,89	3600,67
20	2	14	329	79490674	96,09	571,17	624,30	8,51	555,45	11,03	533,17	3067,90	3601,07
	3	15	377	46897644	93,92	577,00	627,13	7,99	554,36	11,60	659,15	2945,29	3604,44
	4	16	352	59469234	97,54	591,85	616,31	3,97	572,03	7,18	410,39	3190,24	3600,63
	5	14	301	48683271	96,03	517,30	540,75	4,34	497,31	8,03	820,39	2780,15	3600,54
<b>Média</b>	-	-	-	-	85,67	-	-	2,60	-	3,05	-	-	-

**Tabela A.8:** Resultados detalhados do **CGV** para as Classes 2-5 com as *Características Tr* no caso irrestrito.

I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	35	590326	99,44	293,97	317,96	7,54	284,42	10,55	0,63	8,17	8,80
	3	3	13	54954	94,78	284,52	284,52	0,00	284,52	<b>0,00</b>	0,42	0,23	0,65
	4	4	12	17335	91,73	296,75	296,75	0,00	282,95	4,65	0,18	0,05	0,23
	5	4	13	12810	86,15	275,03	285,93	3,81	278,73	2,52	0,24	0,12	0,36
2	2	5	2	2241	78,71	327,44	342,00	4,26	334,96	2,06	0,03	0,05	0,08
	3	5	3	3265	84,96	343,47	353,06	2,71	352,16	0,25	0,03	0,13	0,16



I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	5	5	2835	83,14	327,44	342,00	4,26	334,96	2,06	0,02	0,10	0,12
	5	5	5	3255	85,25	327,44	334,96	2,25	334,96	<b>0,00</b>	0,02	0,07	0,09
3	2	5	22	34920	87,46	394,16	400,15	1,50	387,70	3,11	0,41	0,13	0,54
	3	5	13	25725	84,94	407,45	407,45	0,00	394,72	3,12	0,48	0,16	0,64
	4	5	35	64683	90,65	361,41	374,11	3,39	364,45	2,58	0,49	0,44	0,93
	5	5	40	71316	90,02	354,17	367,26	3,56	358,40	2,41	0,49	0,58	1,07
4	2	6	14	17334	86,78	440,68	440,68	0,00	430,88	2,22	0,17	0,08	0,25
	3	6	16	21157	87,60	442,20	446,61	0,99	430,89	3,52	0,18	0,19	0,37
	4	6	14	32845	92,65	447,37	447,37	0,00	447,37	<b>0,00</b>	0,29	0,23	0,52
	5	6	23	26426	91,29	430,89	430,89	0,00	430,88	<b>0,00</b>	0,20	0,15	0,35
5	2	4	77	420784	96,96	379,94	379,94	0,00	375,28	1,23	1,37	1,91	3,28
	3	4	54	487504	95,80	389,96	392,90	0,75	381,69	2,85	2,48	5,67	8,15
	4	4	50	145884	89,99	383,53	383,88	0,09	383,87	<b>0,00</b>	1,31	1,31	2,62
	5	4	79	265594	95,48	375,28	375,28	0,00	375,28	<b>0,00</b>	1,14	1,42	2,56
6	2	6	35	62729	93,05	492,02	498,16	1,23	495,85	0,46	0,30	0,36	0,66
	3	6	34	97176	95,35	499,97	507,96	1,57	498,16	1,93	0,48	0,63	1,11
	4	6	28	48613	94,16	503,68	505,38	0,34	498,32	1,40	0,37	0,23	0,60
	5	6	31	87370	95,99	482,01	498,16	3,24	495,85	0,46	0,18	0,56	0,74
7	2	5	82	732863	96,64	733,57	751,96	2,45	725,46	3,52	3,51	4,06	7,57
	3	5	41	274279	90,82	699,86	706,99	1,01	678,75	3,99	3,86	2,19	6,05
	4	5	104	380115	93,87	694,53	703,85	1,32	702,45	0,20	4,68	1,80	6,48
	5	4	57	385568	89,92	660,17	661,22	0,16	657,77	0,52	8,75	4,38	13,13
8	2	5	115	660645	95,49	696,00	721,09	3,48	674,55	6,45	3,86	3,92	7,78
	3	5	45	332285	94,92	755,62	771,29	2,03	738,43	4,26	2,71	1,72	4,43
	4	5	47	361985	92,00	706,54	735,28	3,91	692,47	5,82	5,78	3,27	9,05
	5	4	125	837858	89,55	609,55	647,28	5,83	609,90	5,77	20,83	21,34	42,17
9	2	8	29	128647	95,43	615,32	620,46	0,83	607,65	2,06	0,57	1,37	1,94
	3	8	41	460484	98,51	621,97	635,10	2,07	607,65	4,32	1,27	7,90	9,17
	4	8	36	176103	96,76	615,24	633,40	2,87	625,10	1,31	0,71	2,04	2,75
	5	8	47	111333	95,27	607,65	607,65	0,00	607,65	<b>0,00</b>	0,47	0,73	1,20
10	2	6	205	9053351	98,94	694,14	707,50	1,89	689,68	2,52	16,70	96,47	113,17

I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	3	5	190	20683276	99,24	630,70	666,91	5,43	615,68	7,68	33,64	660,82	694,46
	4	7	126	5569282	98,88	743,79	802,42	7,31	711,01	11,39	12,57	83,37	95,94
	5	6	219	17709332	99,44	649,10	704,28	7,83	686,78	2,48	31,45	429,48	460,93
11	2	6	138	3384013	98,14	689,61	743,38	7,23	693,45	6,72	8,79	34,56	43,35
	3	7	120	4864324	99,12	726,31	748,34	2,94	706,73	5,56	8,30	53,88	62,18
	4	7	93	2815838	98,79	827,02	849,50	2,65	786,85	7,37	6,94	22,68	29,62
	5	6	288	36867175	99,67	639,43	658,89	2,95	636,77	3,36	51,46	1489,82	1541,28
12	2	9	67	636831	98,56	623,20	633,20	1,58	610,57	3,57	0,91	8,12	9,03
	3	9	45	484535	97,82	603,98	610,00	0,99	610,00	<b>0,00</b>	0,69	10,26	10,95
	4	9	50	240884	96,84	612,79	614,24	0,24	614,24	<b>0,00</b>	0,76	2,44	3,20
	5	9	34	419070	97,51	604,87	610,23	0,88	610,23	<b>0,00</b>	1,05	10,95	12,00
13	2	7	191	13681393	98,51	2652,89	2783,65	4,70	2585,72	7,11	33,47	309,05	342,52
	3	7	202	15676269	97,27	2540,79	2619,00	2,99	2454,37	6,29	86,97	829,85	916,82
	4	7	174	6852081	97,17	2640,46	2734,91	3,45	2587,63	5,39	44,09	173,94	218,03
	5	6	284	9903960	96,15	2386,93	2464,39	3,14	2334,78	5,26	121,62	767,46	889,08
14	2	6	286	34768653	98,96	1076,49	1185,27	9,18	1038,09	12,42	93,20	3507,33	3600,53
	3	6	134	26446946	98,49	1076,64	1141,32	5,67	996,25	12,71	111,74	3488,78	3600,52
	4	6	183	41445014	99,38	1017,40	1113,96	8,67	981,90	11,86	85,63	3514,92	3600,55
	5	5	323	36515282	99,46	938,50	1011,33	7,20	921,45	8,89	99,98	3500,41	3600,39
15	2	6	186	39782259	99,26	1076,91	1121,31	3,96	1013,29	9,63	81,25	3519,11	3600,36
	3	6	146	42753575	99,25	1166,32	1205,50	3,25	1173,83	2,63	99,10	3501,22	3600,32
	4	7	204	42838223	99,58	1253,83	1311,49	4,40	1234,14	5,90	73,44	3527,00	3600,44
	5	7	215	28597543	99,26	1204,80	1286,01	6,32	1176,68	8,50	79,02	3521,46	3600,48
16	2	11	44	79854	89,77	695,51	698,61	0,44	698,61	<b>0,00</b>	0,42	0,53	0,95
	3	11	45	112882	92,37	695,45	698,61	0,45	698,61	<b>0,00</b>	0,41	0,82	1,23
	4	11	20	72067	89,51	707,58	709,27	0,24	703,35	0,83	0,61	0,51	1,12
	5	11	50	76690	88,28	695,33	698,61	0,47	698,61	<b>0,00</b>	0,39	0,51	0,90
17	2	14	35	259511	95,38	870,13	870,86	0,08	863,66	0,83	0,68	4,39	5,07
	3	14	52	389521	96,86	861,09	861,79	0,08	861,79	<b>0,00</b>	0,39	6,72	7,11
	4	14	55	342944	96,74	861,68	861,79	0,01	861,79	<b>0,00</b>	0,63	4,70	5,33
	5	14	54	398783	97,07	861,09	861,79	0,08	861,79	<b>0,00</b>	0,50	6,58	7,08

I	Cl	#VU	#it <sub>CG</sub>	#verif	arv(%)	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
18	2	9	328	69692740	98,68	1034,49	1078,94	4,12	1034,49	4,12	204,70	3395,65	3600,35
	3	10	321	51528730	98,79	1080,53	1129,27	4,32	1080,53	4,32	165,78	3434,54	3600,32
	4	10	291	53860918	98,65	1141,28	1175,91	2,95	1141,28	2,95	205,21	3395,21	3600,42
	5	8	352	37877250	97,40	954,46	983,60	2,96	944,97	3,93	340,61	3259,76	3600,37
19	2	11	285	29302353	97,44	790,00	820,41	3,71	784,06	4,43	136,01	621,02	757,03
	3	11	207	22757825	96,58	796,00	828,96	3,98	796,00	3,98	167,28	644,65	811,93
	4	12	201	20549489	96,98	817,43	841,96	2,91	809,68	3,83	154,52	629,02	783,54
	5	9	359	61362197	98,45	661,44	682,48	3,08	661,44	3,08	389,81	3210,63	3600,44
20	2	14	330	36504632	97,02	554,37	606,78	8,64	552,98	8,87	212,18	3388,98	3601,16
	3	15	206	24838893	95,71	567,25	608,44	6,77	548,36	9,87	258,29	3342,85	3601,14
	4	16	238	48056904	98,19	583,23	610,66	4,49	574,34	5,95	255,57	3345,01	3600,58
	5	14	395	58680483	98,28	491,53	514,79	4,52	480,10	6,74	432,60	3168,58	3601,18
<b>Média</b>	-	-	-	-	94,82	-	-	2,83	-	3,71	-	-	-

Tabela A.9: Resultados detalhados do CGN para as Classes 2-5 com as Características Tr no caso sequencial.

I	Cl	#VU	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	3	68	2	273,47	290,84	5,97	291,49	<b>-0,22</b>	0,02	4,69	4,71
	3	3	34	124	36	282,70	321,39	12,04	306,03	4,78	11,09	57,63	68,72
	4	4	2	46	1	284,04	294,25	3,47	296,75	<b>-0,85</b>	0,32	0,26	0,58
	5	4	3	36	3	272,95	285,93	4,54	284,23	0,60	4,34	1,03	5,37
2	2	5	9	30	8	337,26	354,01	4,73	347,73	1,77	0,01	1,10	1,11
	3	5	4	22	3	340,72	356,24	4,36	356,24	<b>0,00</b>	0,04	0,55	0,59
	4	5	1	4	0	327,44	342,00	4,26	342,00	<b>0,00</b>	0,00	0,08	0,08
	5	5	1	3	0	327,44	334,96	2,25	334,96	<b>0,00</b>	0,00	0,11	0,11
3	2	5	20	119	35	368,07	403,93	8,88	403,93	<b>0,00</b>	0,08	10,41	10,49
	3	5	2	72	3	378,45	394,72	4,12	394,72	<b>0,00</b>	1,39	0,57	1,96
	4	5	4	71	5	356,82	382,85	6,80	372,56	2,69	1,42	1,91	3,33
	5	4	2	48	1	351,82	376,18	6,48	358,40	4,73	1,06	1,21	2,27

I	Cl	#VU	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
4	2	6	5	46	4	436,98	441,97	1,13	440,94	0,23	0,03	0,87	0,90
	3	6	2	36	1	436,98	441,97	1,13	445,25	<b>-0,74</b>	0,71	0,40	1,11
	4	6	1	17	0	446,09	447,37	0,29	447,37	<b>0,00</b>	0,00	0,62	0,62
	5	6	1	23	0	430,89	430,89	0,00	430,88	<b>0,00</b>	0,00	0,09	0,09
5	2	4	6	128	9	383,25	390,62	1,89	388,72	0,49	0,31	4,24	4,55
	3	4	3	114	2	381,69	381,69	0,00	383,15	<b>-0,38</b>	0,09	3,52	3,61
	4	4	1	76	0	382,28	383,88	0,42	385,70	<b>-0,48</b>	0,16	0,85	1,01
	5	4	1	96	0	374,13	375,28	0,31	375,28	<b>0,00</b>	0,00	0,48	0,48
6	2	6	1	31	0	482,09	499,08	3,40	499,08	<b>0,00</b>	0,00	0,32	0,32
	3	6	4	68	3	483,47	504,68	4,20	504,68	<b>0,00</b>	0,69	1,17	1,86
	4	6	1	47	0	491,11	498,65	1,51	501,04	<b>-0,48</b>	0,61	0,73	1,34
	5	6	1	19	0	482,01	500,59	3,71	495,85	0,95	8,30	0,52	8,82
7	2	5	5	137	4	684,32	756,29	9,52	734,65	2,86	0,10	16,33	16,43
	3	5	11	172	21	676,63	713,94	5,23	702,59	1,59	3,95	67,37	71,32
	4	5	3	132	3	682,05	707,26	3,57	703,64	0,51	5,44	7,16	12,60
	5	4	5	134	4	645,22	661,22	2,42	669,17	<b>-1,20</b>	20,76	15,65	36,41
8	2	5	19	186	32	663,23	731,32	9,31	727,40	0,54	0,14	148,97	149,11
	3	5	11	141	20	715,76	745,32	3,97	741,12	0,56	4,36	21,62	25,98
	4	4	5	152	9	672,64	708,23	5,02	723,55	<b>-2,16</b>	179,86	22,05	201,91
	5	4	21	256	27	602,19	646,73	6,89	646,46	0,04	147,42	408,11	555,53
9	2	8	2	50	1	611,49	611,49	0,00	611,49	<b>0,00</b>	0,00	0,55	0,55
	3	8	3	52	2	613,90	613,90	0,00	622,15	<b>-1,34</b>	1,48	1,23	2,71
	4	8	4	49	3	614,57	632,14	2,78	625,13	1,11	0,36	3,83	4,19
	5	8	1	25	0	607,65	607,65	0,00	607,65	<b>0,00</b>	0,00	0,19	0,19
10	2	6	53	484	132	631,46	795,48	20,62	700,20	11,98	5,98	6403,09	6409,07
	3	5	10	430	15	579,80	674,96	14,10	636,80	5,65	6,48	7457,06	7463,54
	4	7	26	355	48	689,17	804,60	14,35	763,98	5,05	52,72	1550,80	1603,52
	5	6	6	313	9	618,36	707,68	12,62	705,62	0,29	84,66	1020,15	1104,81
11	2	6	14	356	36	620,93	768,90	19,24	722,44	6,04	0,40	1595,35	1595,75
	3	7	25	311	47	667,77	771,27	13,42	730,30	5,31	5,42	790,57	795,99
	4	7	116	580	289	768,51	901,71	14,77	832,91	7,63	91,11	1615,32	1706,43

I	Cl	#VU	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	5	6	14	387	27	611,85	694,00	11,84	649,75	6,38	575,22	4749,09	5324,31
12	2	9	7	68	12	609,90	619,63	1,57	620,47	<b>-0,14</b>	0,02	38,87	38,89
	3	9	1	35	0	603,23	615,41	1,98	610,00	0,88	0,00	7,40	7,40
	4	9	2	45	2	608,40	623,79	2,47	623,71	0,01	2,33	5,38	7,71
	5	9	1	44	0	603,23	614,94	1,91	610,23	0,77	0,00	9,24	9,24
13	2	6	15	349	45	2414,85	-	-	2669,39	-	0,26	14400,17	14400,43
	3	5	16	478	69	2388,77	-	-	2500,55	-	17,48	14382,91	14400,39
	4	7	26	418	59	2497,07	2725,53	8,38	2703,48	0,81	107,15	6962,53	7069,68
	5	6	16	556	42	2357,41	2511,29	6,13	2490,87	0,81	790,11	9555,50	10345,61
14	2	5	1	399	4	1002,69	-	-	1139,70	-	0,00	14400,59	14400,59
	3	5	1	362	4	1004,96	-	-	1101,96	-	0,01	14400,44	14400,45
	4	5	1	472	3	930,78	-	-	1000,76	-	0,00	14400,78	14400,78
	5	5	1	391	1	948,80	-	-	959,51	-	0,00	14400,61	14400,61
15	2	5	1	400	4	992,83	-	-	1116,51	-	0,00	14400,31	14400,31
	3	6	1	366	2	1061,21	-	-	1187,59	-	0,00	14400,74	14400,74
	4	7	1	206	4	1166,08	-	-	1288,84	-	0,00	14400,52	14400,52
	5	6	1	262	6	1154,72	-	-	1265,14	-	0,01	14400,35	14400,36
16	2	11	1	49	0	695,33	703,57	1,17	698,60	0,71	0,00	0,83	0,83
	3	11	1	54	0	695,33	698,61	0,47	698,61	<b>0,00</b>	0,00	0,78	0,78
	4	11	4	81	5	701,28	711,43	1,43	709,27	0,30	4,96	1,73	6,69
	5	11	1	38	0	695,33	703,35	1,14	698,60	0,68	0,00	0,75	0,75
17	2	14	3	64	4	870,12	870,86	0,08	870,86	<b>0,00</b>	0,00	11,67	11,67
	3	14	1	61	0	861,09	861,79	0,08	861,79	<b>0,00</b>	0,00	4,03	4,03
	4	14	1	65	0	861,09	861,79	0,08	862,62	<b>-0,10</b>	0,12	5,71	5,83
	5	14	1	56	0	861,09	861,79	0,08	861,79	<b>0,00</b>	0,00	5,84	5,84
<b>Média</b>	-	-	-	-	16,34	-	-	4,87	-	1,18	-	-	-

**Tabela A.10:** Resultados detalhados do CGN para as Classes 2-5 com as *Características Tr* no caso irrestrito.

I	Cl	#VU	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	1	31	0	273,47	294,27	7,07	284,42	3,35	0,04	1,66	1,70
	3	3	2	46	1	276,09	284,52	2,96	284,52	<b>0,00</b>	25,23	4,55	29,78
	4	4	1	27	0	282,95	282,95	0,00	282,95	<b>0,00</b>	0,00	0,08	0,08
	5	4	2	35	1	272,16	285,93	4,82	278,73	2,52	29,16	0,44	29,60
2	2	5	1	2	0	327,44	342,00	4,26	334,96	2,06	0,00	0,09	0,09
	3	5	2	10	1	340,72	353,06	3,49	352,16	0,25	0,30	0,31	0,61
	4	5	1	5	0	327,44	342,00	4,26	334,96	<b>2,06</b>	0,00	0,37	0,37
	5	5	1	5	0	327,44	334,96	2,25	334,96	<b>0,00</b>	0,00	0,09	0,09
3	2	5	8	79	12	367,95	393,54	6,50	387,70	1,48	0,08	4,08	4,16
	3	5	2	53	2	378,45	394,72	4,12	394,72	<b>0,00</b>	14,29	0,95	15,24
	4	4	1	46	0	352,92	372,23	5,19	364,45	2,09	121,19	0,63	121,82
	5	4	1	45	0	351,74	368,34	4,51	358,40	2,70	0,00	0,51	0,51
4	2	6	1	26	0	430,89	430,89	0,00	430,88	<b>0,00</b>	0,00	0,25	0,25
	3	6	1	24	0	430,89	430,89	0,00	430,89	<b>0,00</b>	0,73	0,19	0,92
	4	6	1	19	0	446,09	447,37	0,29	447,37	<b>0,00</b>	0,00	0,64	0,64
	5	6	1	23	0	430,89	430,89	0,00	430,88	<b>0,00</b>	0,00	0,16	0,16
5	2	4	1	62	0	374,13	375,28	0,31	375,28	<b>0,00</b>	0,04	0,52	0,56
	3	4	3	109	2	381,69	381,69	0,00	381,69	<b>0,00</b>	0,34	4,10	4,44
	4	4	1	64	0	382,28	383,88	0,42	383,87	<b>0,00</b>	0,01	1,06	1,07
	5	4	1	98	0	374,13	375,28	0,31	375,28	<b>0,00</b>	0,00	1,05	1,05
6	2	6	1	70	0	482,09	499,08	3,40	495,85	0,65	0,00	0,72	0,72
	3	6	2	55	1	483,47	500,43	3,39	498,16	<b>0,45</b>	0,22	0,87	1,09
	4	6	1	56	0	491,11	498,65	1,51	498,32	<b>0,07</b>	0,07	0,84	0,91
	5	6	1	31	0	482,01	498,16	3,24	495,85	0,46	0,00	0,58	0,58
7	2	5	3	115	2	689,44	725,46	4,96	725,46	0,00	0,04	11,04	11,08
	3	4	5	113	6	666,02	678,75	1,88	678,75	<b>0,00</b>	28,54	34,61	63,15
	4	5	3	142	2	682,05	703,85	3,10	702,45	0,20	21,27	14,16	35,43
	5	4	1	114	0	644,75	657,77	1,98	657,77	<b>0,00</b>	3,29	4,76	8,05
8	2	4	7	170	8	657,49	679,18	3,19	674,55	0,68	0,41	51,45	51,86
	3	5	2	100	1	712,32	740,85	3,85	738,43	<b>0,33</b>	2,72	5,39	8,11

I	CI	#VU	#it <sub>EL</sub>	#it <sub>CG</sub>	#Cols <sub>rej</sub>	RL	LS	gap <sub>RL</sub> (%)	lit*	gap <sub>lit*</sub> (%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	4	1	123	0	670,00	692,47	3,25	692,47	<b>0,00</b>	4,27	12,50	16,77
	5	3	1	241	0	597,27	613,01	2,57	609,90	<b>0,51</b>	56,68	46,40	103,08
9	2	8	1	27	0	607,65	607,65	0,00	607,65	<b>0,00</b>	0,00	0,27	0,27
	3	8	1	37	0	607,65	607,65	0,00	607,65	<b>0,00</b>	0,01	4,78	4,79
	4	8	3	73	2	614,57	632,46	2,83	625,10	1,16	11,00	3,35	14,35
	5	8	1	45	0	607,65	607,65	0,00	607,65	<b>0,00</b>	0,00	0,82	0,82
10	2	6	26	447	68	626,19	748,43	16,33	689,68	7,85	90,68	3886,85	3977,53
	3	5	4	404	6	579,21	621,52	6,81	615,68	<b>0,94</b>	32,55	4961,81	4994,36
	4	6	2	307	1	683,66	713,93	4,24	711,01	<b>0,41</b>	21,45	564,85	586,30
	5	6	4	341	3	618,22	699,69	11,64	686,78	1,84	1553,34	1039,18	2592,52
11	2	6	14	379	35	624,68	743,38	15,97	693,45	6,72	15,27	1941,52	1956,79
	3	7	21	258	30	673,13	746,64	9,84	706,73	5,34	78,50	704,07	782,57
	4	7	9	259	16	743,56	792,58	6,18	786,85	<b>0,72</b>	81,53	203,21	284,74
	5	6	1	339	0	610,27	624,89	2,34	636,77	<b>-1,90</b>	72,94	2392,74	2465,68
12	2	9	6	86	5	605,74	623,53	2,85	610,57	2,08	0,04	31,40	31,44
	3	9	1	49	0	603,23	610,00	1,11	610,00	<b>0,00</b>	0,00	17,20	17,20
	4	9	2	55	1	606,40	619,91	2,18	614,24	0,91	0,86	6,95	7,81
	5	9	1	36	0	603,23	610,23	1,15	610,23	<b>0,00</b>	0,00	10,78	10,78
13	2	6	8	464	15	2409,21	2641,78	8,80	2585,72	2,12	0,62	8252,85	8253,47
	3	6	16	450	38	2396,70	2508,53	4,46	2454,37	2,16	398,10	12049,78	12447,88
	4	7	12	315	19	2488,95	2734,91	8,99	2587,63	5,39	2343,57	2254,73	4598,30
	5	5	3	397	4	2331,68	2334,78	0,13	2334,78	<b>0,00</b>	819,88	1442,37	2262,25
14	2	5	1	383	3	992,18	-	-	1038,09	-	0,01	14400,50	14400,51
	3	5	1	331	4	994,23	-	-	996,25	-	0,00	14400,59	14400,59
	4	5	1	399	2	941,30	-	-	981,90	-	0,00	14400,44	14400,44
	5	5	1	496	1	910,14	-	-	921,45	-	0,00	14400,51	14400,51
15	2	5	1	331	3	977,26	-	-	1013,29	-	0,00	14400,42	14400,42
	3	6	1	253	2	1068,79	-	-	1173,83	-	0,00	14400,63	14400,63
	4	7	1	265	2	1171,73	-	-	1234,14	-	0,00	14400,55	14400,55
	5	6	1	206	4	1161,80	-	-	1176,68	-	0,00	14400,28	14400,28
16	2	11	1	60	0	695,33	698,61	0,47	698,61	<b>0,00</b>	0,00	0,88	0,88

<b>I</b>	<b>Cl</b>	<b>#VU</b>	<b>#it<sub>EL</sub></b>	<b>#it<sub>CG</sub></b>	<b>#Cols<sub>rej</sub></b>	<b>RL</b>	<b>LS</b>	<b>gap<sub>RL</sub>(%)</b>	<b>lit*</b>	<b>gap<sub>lit*</sub>(%)</b>	<b>T<sub>car</sub>(s)</b>	<b>T<sub>rot</sub>(s)</b>	<b>T<sub>tot</sub>(s)</b>
	3	11	1	52	0	695,33	698,61	0,47	698,61	<b>0,00</b>	0,00	1,06	1,06
	4	11	3	50	2	695,97	704,66	1,23	703,35	<b>0,19</b>	2,70	1,81	4,51
	5	11	1	50	0	695,33	698,61	0,47	698,61	<b>0,00</b>	0,00	0,60	0,60
17	2	14	3	51	2	870,03	870,86	0,10	863,66	<b>0,83</b>	0,03	11,59	11,62
	3	14	1	52	0	861,09	861,79	0,08	861,79	<b>0,00</b>	0,00	6,66	6,66
	4	14	1	53	0	861,09	861,79	0,08	861,79	<b>0,00</b>	0,00	3,85	3,85
	5	14	1	54	0	861,09	861,79	0,08	861,79	<b>0,00</b>	0,00	6,58	6,58
<b>Média</b>	-	-	-	-	4,51	-	-	3,27	-	0,94	-	-	-



As Tabelas A.11-A.13 apresentam os resultados obtidos pelo BC desenvolvido. Tais Tabelas apresentam oito colunas em comum, que são citadas a seguir. A coluna **I** apresenta o número da instância analisada. A coluna **#VU** exibe o número de veículos utilizados pela solução obtida. A coluna **#it<sub>BC</sub>** apresenta o número de iterações realizadas pelo BC, ou seja, o número de soluções básicas analisadas na resolução das relaxações dos nós da árvore de BB. A coluna **#CCCs** indica o número de CCCs inseridas no modelo. As colunas **LI** e **LS** apresentam, respectivamente, o limite inferior e superior obtidos. As soluções ótimas encontradas foram destacadas em negrito na coluna **LS**. A coluna **gap(%)** apresenta o *gap* percentual entre o limite superior e o limite inferior obtidos, por meio da equação  $100 \times (LS - LI)/LS$ . Por fim, a coluna **T<sub>tot</sub>(s)** exibe o tempo de execução total do algoritmo. Nas Tabelas A.12-A.13, outras seis colunas também são consideradas. A coluna **CI** indica qual a classe da instância. A coluna **#IPCs** apresenta o número de IPCs inseridas no modelo, ou seja, o número de rotas inviáveis detectadas. A coluna **#Via** exibe o número de rotas viáveis analisadas. A coluna **#FA** apresenta se ocorreu alguma falha (1) ou não (0) durante a verificação da viabilidade. Uma falha acontece quando o algoritmo BB, responsável por verificar a viabilidade dos carregamentos, não consegue obter um resultado durante o seu limite de tempo de execução, sendo forçado a considerar o carregamento inviável sem garantias de estar correto. As colunas **T<sub>car</sub>(s)** e **T<sub>rot</sub>(s)** apresentam o tempo gastos para resolver operações de carregamento e roteamento, respectivamente. Em alguns casos, o CPLEX teve sua execução interrompida por falta de memória no computador utilizado. Nesses casos, o **T<sub>tot</sub>(s)** exibe um sinal “>” à esquerda do tempo, pois o tempo considerado foi medido antes do algoritmo ter a execução suspensa. Por consequência, algumas colunas tiveram que ser preenchidas com “-”, pois nem todos os dados puderam ser obtidos.

**Tabela A.11:** Resultados detalhados do BC para a Classe 1 com as *Características Tr*.

<b>I</b>	<b>#VU</b>	<b>#it<sub>BC</sub></b>	<b>#CCCs</b>	<b>LI</b>	<b>LS</b>	<b>gap(%)</b>	<b>T<sub>tot</sub>(s)</b>
1	3	883	104	278,73	<b>278,73</b>	0,00	0,17
2	5	4627	378	334,96	<b>334,96</b>	0,00	0,39
3	4	2790	266	358,40	<b>358,40</b>	0,00	0,38
4	6	25761	720	430,85	<b>430,89</b>	0,01	1,84
5	4	295	70	375,28	<b>375,28</b>	0,00	0,10
6	6	9816	670	495,83	<b>495,85</b>	0,00	1,07
7	3	45	20	568,56	<b>568,56</b>	0,00	0,04
8	3	45	23	568,56	<b>568,56</b>	0,00	0,04
9	8	60776	860	607,61	<b>607,65</b>	0,01	4,06
10	3	20125	516	535,77	<b>535,80</b>	0,01	2,28
11	4	753	100	505,01	<b>505,01</b>	0,00	0,23
12	-	-	-	570,92	-	-	>7947,57
13	3	4382	208	2006,34	<b>2006,34</b>	0,00	0,48
14	4	22821	1115	837,63	<b>837,67</b>	0,01	4,43
15	4	19429	1072	837,66	<b>837,67</b>	0,00	3,55
16	-	-	-	682,83	702,11	2,75	>18811,03
17	-	-	-	783,63	-	-	>9171,73

Tabela A.12: Resultados detalhados do BC para as Classes 2-5 com as *Características Tr* no caso sequencial.

I	Cl	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	52136	1197	118	155	0	290,82	<b>290,84</b>	0,01	32,51	4,50	37,01
	3	3	4259	263	7	11	0	284,52	<b>284,52</b>	0,00	15,60	0,42	16,02
	4	4	7960	374	6	22	0	294,25	<b>294,25</b>	0,00	3,77	0,55	4,32
	5	3	2008	146	5	13	1	278,73	278,73	0,00	26923,40	0,24	26923,64
2	2	5	39954	871	12	78	0	347,73	<b>347,73</b>	0,00	0,04	2,26	2,30
	3	5	39681	649	8	57	0	352,13	<b>352,16</b>	0,01	1,34	1,97	3,31
	4	5	11124	510	2	23	0	341,98	<b>342,00</b>	0,01	36,21	0,75	36,96
	5	5	4627	378	0	5	0	334,96	<b>334,96</b>	0,00	0,00	0,40	0,40
3	2	5	38177	911	8	6	0	373,39	<b>373,43</b>	0,01	0,09	3,52	3,61
	3	5	131225	2374	46	113	0	394,69	<b>394,72</b>	0,01	132,19	29,32	161,51
	4	4	31244	1152	27	93	0	370,79	<b>370,82</b>	0,01	972,68	4,17	976,85
	5	4	2654	285	0	4	0	358,40	<b>358,40</b>	0,00	0,00	0,36	0,36
4	2	6	119724	1440	11	85	0	440,90	<b>440,94</b>	0,01	0,28	13,01	13,29
	3	6	54609	1510	5	37	0	440,64	<b>440,68</b>	0,01	9,87	6,98	16,85
	4	6	52491	1018	0	30	0	447,34	<b>447,37</b>	0,01	0,01	4,79	4,80
	5	6	57799	1940	0	12	0	430,85	<b>430,89</b>	0,01	0,02	9,47	9,49
5	2	4	2372	157	18	38	0	388,72	<b>388,72</b>	0,00	3,23	0,26	3,49
	3	4	723	102	4	28	0	381,69	<b>381,69</b>	0,00	0,36	0,16	0,52
	4	4	999	122	4	16	0	383,88	<b>383,88</b>	0,00	9,92	0,19	10,11
	5	4	295	70	0	12	0	375,28	<b>375,28</b>	0,00	0,10	0,08	0,18
6	2	6	18518	918	12	78	0	499,04	<b>499,08</b>	0,01	0,14	2,18	2,32
	3	6	36728	1074	16	116	0	504,63	<b>504,68</b>	0,01	17,02	4,32	21,34
	4	6	14355	1062	2	28	0	498,29	<b>498,32</b>	0,01	297,95	1,79	299,74
	5	6	5723	537	0	48	0	495,84	<b>495,85</b>	0,00	0,00	0,67	0,67
7	2	5	78234	1185	22	31	0	713,83	<b>713,89</b>	0,01	1,35	9,37	10,72
	3	4	329129	1261	121	96	0	702,52	<b>702,59</b>	0,01	566,72	38,55	605,27
	4	4	58096	1020	47	119	0	703,45	<b>703,49</b>	0,01	28151,62	6,32	28157,94
	5	4	65499	1077	26	54	1	652,14	669,53	2,60	86399,09	6,33	86405,42
8	2	5	212040	1113	223	167	0	705,84	<b>705,90</b>	0,01	6,42	23,11	29,53
	3	5	102240	1080	29	40	0	740,17	<b>740,24</b>	0,01	63,92	11,31	75,23

I	CI	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	4	18300	509	14	42	0	697,92	<b>697,92</b>	0,00	1500,56	1,49	1502,05
	5	4	78503	544	255	262	1	635,19	646,60	1,77	86399,82	5,59	86405,41
9	2	8	128631	2328	0	8	0	607,60	<b>607,65</b>	0,01	0,00	22,08	22,08
	3	8	188533	3111	0	8	0	607,60	<b>607,65</b>	0,01	0,00	46,04	46,04
	4	-	-	-	-	-	-	-	-	-	-	-	-
	5	8	142726	2542	0	8	0	607,65	<b>607,65</b>	0,00	0,00	28,31	28,31
10	2	-	-	-	-	-	-	676,51	740,41	8,63	-	-	>10242,77
	3	5	269951	580	67	10	0	616,26	<b>616,32</b>	0,01	151,32	14,62	165,94
	4	6	105689	1748	159	285	1	712,04	728,05	2,20	86331,45	102,95	86434,40
	5	5	38465	1681	24	36	1	638,86	723,49	11,70	86324,45	81,99	86406,44
11	2	-	-	-	-	-	-	669,26	697,71	4,08	-	-	>9265,54
	3	6	6789504	5429	958	1581	0	717,30	<b>717,37</b>	0,01	6416,74	5542,71	11959,45
	4	7	16393646	2858	72	33	0	777,71	<b>777,79</b>	0,01	2552,69	5257,09	7809,78
	5	-	-	-	-	-	-	615,97	-	-	-	-	86400,01
12	2	-	-	-	-	-	-	573,57	-	-	-	-	>13914,38
	3	-	-	-	-	-	-	576,27	-	-	-	-	>12277,41
	4	-	-	-	-	-	-	578,10	-	-	-	-	>13371,45
	5	-	-	-	-	-	-	581,29	641,15	9,34	-	-	>11537,85
13	2	-	-	-	-	-	-	2468,12	2525,91	2,29	-	-	>8849,35
	3	-	-	-	-	-	-	2400,31	2551,04	5,91	-	-	>10544,95
	4	6	22145375	4115	199	353	0	2609,10	<b>2609,36</b>	0,01	47272,93	12067,08	59340,01
	5	-	-	1691	-	-	-	2288,86	-	-	-	-	86400,25
14	2	-	-	-	-	-	-	1010,01	1121,33	9,93	-	-	>16757,77
	3	-	-	-	-	-	-	969,23	1210,27	19,92	-	-	14348,67
	4	-	-	-	-	-	-	931,94	1029,85	9,51	-	-	>57978,99
	5	-	-	-	-	-	-	851,95	-	-	-	-	86400,83
15	2	-	-	-	-	-	-	983,39	1125,69	12,64	-	-	14310,85
	3	-	-	-	-	-	-	-	-	-	-	-	-
	4	-	-	-	-	-	-	1140,12	1278,22	10,80	-	-	>68071,33
	5	-	-	-	-	-	-	1114,76	-	-	-	-	86433,9
16	2	11	22307652	13344	1	98	0	698,54	<b>698,61</b>	0,01	0,84	38472,41	38473,25

I	Cl	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	3	-	-	-	-	-	-	683,15	705,06	3,11	-	-	>9509,94
	4	-	-	-	-	-	-	665,38	776,00	14,26	-	-	>17372,64
	5	-	-	-	-	-	-	672,07	709,59	5,29	-	-	>16181,59
17	2	-	-	-	-	-	-	784,33	-	-	-	-	>14426,69
	3	-	-	-	-	-	-	781,64	-	-	-	-	>18297,46
	4	-	-	-	-	-	-	779,44	-	-	-	-	>9220,42
	5	-	-	-	-	-	-	782,33	-	-	-	-	>11680,93

**Tabela A.13:** Resultados detalhados do BC para as Classes 2-5 com as *Características Tr* no caso irrestrito.

I	Cl	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
1	2	3	5870	422	10	26	0	278,73	<b>278,73</b>	0,00	58,97	0.54	59,51
	3	3	3720	266	6	21	0	284,52	<b>284,52</b>	0,00	232,86	0.34	233,20
	4	4	3657	325	2	26	0	282,94	<b>282,95</b>	0,00	36,95	0.36	37,31
	5	3	1051	106	0	12	0	278,73	<b>278,73</b>	0,00	10,17	0.16	10,33
2	2	5	6817	467	0	25	0	334,96	<b>334,96</b>	0,00	0,00	0.52	0,52
	3	5	28992	689	3	47	0	352,14	<b>352,16</b>	0,00	3,61	1.63	5,24
	4	5	4644	372	0	10	0	334,96	<b>334,96</b>	0,00	5,57	0.39	5,96
	5	5	4627	378	0	5	0	334,96	<b>334,96</b>	0,00	0,00	0.4	0,40
3	2	5	35229	1143	18	31	0	373,29	<b>373,29</b>	0,00	0,29	4.17	4,46
	3	5	70755	1322	15	32	0	390,51	<b>390,55</b>	0,01	3142,02	8.41	3150,43
	4	4	5164	455	1	16	0	362,41	<b>362,41</b>	0,00	1110,77	0.62	1111,39
	5	4	2654	285	0	4	0	358,40	<b>358,40</b>	0,00	0,00	0.36	0,36
4	2	6	17581	635	0	18	0	430,86	<b>430,89</b>	0,01	0,00	1.2	1,20
	3	6	9954	520	0	6	0	430,86	<b>430,89</b>	0,01	2,47	0.8	3,27
	4	6	52491	1018	0	30	0	447,34	<b>447,37</b>	0,01	0,00	4.83	4,83
	5	6	57799	1940	0	12	0	430,85	<b>430,89</b>	0,01	0,00	9.48	9,48
5	2	4	295	70	0	12	0	375,28	<b>375,28</b>	0,00	0,60	0.08	0,68
	3	4	723	102	4	28	0	381,69	<b>381,69</b>	0,00	1,71	0.16	1,87
	4	4	1118	126	3	25	0	383,88	<b>383,88</b>	0,00	256,42	0.18	256,60

I	CI	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	5	4	295	70	0	12	0	375,28	<b>375,28</b>	0,00	0,00	0.09	0,09
6	2	6	6485	548	1	41	0	495,80	<b>495,85</b>	0,01	0,12	0.72	0,84
	3	6	13282	786	1	47	0	498,16	<b>498,16</b>	0,00	2,61	1.44	4,05
	4	6	14841	964	0	30	0	498,29	<b>498,32</b>	0,01	58,58	1.65	60,23
	5	6	5723	537	0	48	0	495,84	<b>495,85</b>	0,00	0,00	0.69	0,69
7	2	5	19512	725	5	20	0	699,46	<b>699,52</b>	0,01	0,56	1.79	2,35
	3	4	27966	474	76	93	0	678,69	<b>678,75</b>	0,01	2917,69	1.91	2919,60
	4	4	43638	620	26	94	1	700,65	700,72	0,01	16595,40	2.92	16598,32
	5	4	31544	918	5	39	1	657,77	657,77	0,00	21642,14	3.69	21645,83
8	2	4	6865	300	42	114	0	674,51	<b>674,55</b>	0,01	1,95	0.65	2,60
	3	4	105724	1090	28	121	0	738,37	<b>738,43</b>	0,01	1904,00	11.82	1915,82
	4	4	10670	355	2	18	1	692,44	692,47	0,01	7207,24	0.75	7207,99
	5	3	2945	234	9	24	1	609,90	609,90	0,00	21775,22	0.37	21775,59
9	2	8	128631	2328	0	8	0	607,60	<b>607,65</b>	0,01	0,00	23.2	23,20
	3	8	188533	3111	0	8	0	607,60	<b>607,65</b>	0,01	0,00	47.04	47,04
	4	-	-	-	-	-	-	-	-	-	-	-	-
	5	8	142726	2542	0	8	0	607,65	<b>607,65</b>	0,00	0,00	28.87	28,87
10	2	5	8819404	2511	979	1301	0	689,62	<b>689,68</b>	0,01	818,08	3460.41	4278,49
	3	5	218428	583	65	31	0	615,62	<b>615,68</b>	0,01	5797,22	13.93	5811,15
	4	6	32243	1108	23	103	1	710,82	710,87	0,01	58041,43	3.81	58045,24
	5	5	150703	2231	32	146	1	674,05	689,34	2,22	86368,33	37.09	86405,42
11	2	-	-	-	-	-	-	666,76	701,59	4,96	-	-	>5173,11
	3	6	1827648	2550	123	404	1	706,66	706,73	0,01	15898,16	926.31	16824,47
	4	7	15243694	3001	70	53	1	778,02	778,10	0,01	18324,82	5273.21	23598,03
	5	5	16466	710	7	58	1	624,77	624,82	0,01	37508,75	1.73	37510,48
12	2	-	-	-	-	-	-	573,68	-	-	-	-	>13994,42
	3	-	-	-	-	-	-	576,27	-	-	-	-	>12182,78
	4	-	-	-	-	-	-	578,10	-	-	-	-	>13574,36
	5	-	-	-	-	-	-	581,76	636,91	8,66	-	-	>11832,89
13	2	6	33901779	7705	150	29	0	2511,48	<b>2511,73</b>	0,01	20,76	32963.05	32983,81
	3	6	14706061	3639	97	95	0	2436,32	<b>2436,56</b>	0,01	3024,41	6479.3	9503,71

<b>I</b>	<b>CI</b>	#VU	#it <sub>BC</sub>	#CCCs	#IPCs	#Via	#FA	LI	LS	gap(%)	T <sub>car</sub> (s)	T <sub>rot</sub> (s)	T <sub>tot</sub> (s)
	4	7	83440	1704	41	67	1	2483,66	2728,30	8,97	86356,42	49.46	86405,88
	5	5	57023	1312	39	117	1	2314,95	2437,99	5,05	86214,19	207.96	86422,15
14	2	-	-	-	-	-	-	1010,27	1077,41	6,23	-	-	>15001,80
	3	-	-	-	-	-	-	977,54	1012,29	3,43	-	-	>27876,52
	4	5	3737580	28157	29	66	1	934,00	997,30	6,35	68738,54	17697.4	86435,94
	5	5	903254	16970	17	27	1	856,76	973,68	12,01	83224,42	3225.51	86449,93
15	2	-	-	-	-	-	-	982,43	1111,69	11,63	-	-	>10685,53
	3	-	-	-	-	-	-	-	-	-	-	-	-
	4	7	1578957	18868	34	74	1	1138,22	1253,90	9,23	81630,88	4796.41	86427,29
	5	7	1376534	17948	16	43	1	1120,21	1293,41	13,39	82542,20	3888.3	86430,50
16	2	-	-	-	-	-	-	682,11	701,33	2,74	-	-	>12945,43
	3	-	-	-	-	-	-	681,78	707,62	5,42	-	-	>18074,38
	4	-	-	-	-	-	-	670,97	731,87	8,32	-	-	>17660,89
	5	-	-	-	-	-	-	671,14	709,59	5,42	-	-	>18074,38
17	2	-	-	-	-	-	-	783,74	-	-	-	-	>12301,78
	3	-	-	-	-	-	-	780,08	-	-	-	-	>14323,73
	4	-	-	-	-	-	-	781,92	-	-	-	-	16993,82
	5	-	-	-	-	-	-	782,65	-	-	-	-	>13835,02