
Universidade Estadual de Campinas
Instituto de Computação


Mauro Henrique Mulati

# Approaches for Vehicle Routing Problems with Energy Considerations and Selective Backhauls

# Abordagens para Problemas de Roteamento de Veículos com Considerações de Energia e Backhauls Seletivos

CAMPINAS
2022

**Mauro Henrique Mulati**

# Approaches for Vehicle Routing Problems with Energy Considerations and Selective Backhauls

# Abordagens para Problemas de Roteamento de Veículos com Considerações de Energia e Backhauls Seletivos

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

**Supervisor/Orientador: Prof. Dr. Flávio Keidi Miyazawa**
**Co-supervisor/Coorientador: Prof. Dr. Ricardo Fukasawa**

Este exemplar corresponde à versão final da Tese defendida por Mauro Henrique Mulati e orientada pelo Prof. Dr. Flávio Keidi Miyazawa.

CAMPINAS
2022

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M896a

Mulati, Mauro Henrique, 1982-
Approaches for vehicle routing problems with energy considerations and selective backhauls / Mauro Henrique Mulati. – Campinas, SP : [s.n.], 2022.

Orientador: Flávio Keidi Miyazawa.
Coorientador: Ricardo Fukasawa.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Problema de roteamento de veículos. 2. Otimização combinatória. 3. Programação linear inteira. 4. Otimização robusta. 5. Pesquisa operacional. I. Miyazawa, Flávio Keidi, 1970-. II. Fukasawa, Ricardo. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações Complementares

**Título em outro idioma:** Abordagens para problemas de roteamento de veículos com considerações de energia e backhauls seletivos
**Palavras-chave em inglês:**
Vehicle routing problem
Combinatorial optimization
Integer linear programming
Robust optimization
Operations research
**Área de concentração:** Ciência da Computação
**Titulação:** Doutor em Ciência da Computação
**Banca examinadora:**
Flávio Keidi Miyazawa [Orientador]
Eduardo Uchoa Barboza
Rafael Martinelli Pinto
Kelly Cristina Poldi
Reinaldo Morabito Neto
**Data de defesa:** 19-12-2022
**Programa de Pós-Graduação:** Ciência da Computação

**Identificação e informações acadêmicas do(a) aluno(a)**
- ORCID do autor: https://orcid.org/0000-0002-1945-7452
- Currículo Lattes do autor: http://lattes.cnpq.br/3945457468863985

**Universidade Estadual de Campinas**
**Instituto de Computação**


**Mauro Henrique Mulati**

# Approaches for Vehicle Routing Problems with Energy Considerations and Selective Backhauls

# Abordagens para Problemas de Roteamento de Veículos com Considerações de Energia e Backhauls Seletivos

**Banca Examinadora:**

- Prof. Dr. Flávio Keidi Miyazawa
  IC/UNICAMP
- Prof. Dr. Eduardo Uchoa Barboza
  TEP/UFF
- Prof. Dr. Rafael Martinelli Pinto
  DEI/PUC-Rio
- Profa. Dra. Kelly Cristina Poldi
  IMECC/UNICAMP
- Prof. Dr. Reinaldo Morabito Neto
  DEP/UFSCar

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 19 de dezembro de 2022

# Agradecimentos

Agradeço a Deus pelos caminhos e pessoas que me direcionaram até aqui e pela força para concluir este doutorado. Agradeço à minha esposa, Daniele, à nossa filha, Júlia, e ao nosso filho, a caminho, por todo apoio, carinho e paciência durante a realização deste trabalho. Agradeço aos meus pais, Mauro e Marlene, por todo apoio e incentivo na vida e nos estudos. Agradeço ao apoio do meu irmão, Fabio, da minha irmã, Thatiane, dos meus sogros, José e Amélia, e de toda a família. Agradeço o incentivo do meu amigo Rogério.

Agradeço ao Flávio e ao Ricardo pelas valiosas orientações e ensinamentos, fundamentais neste doutorado. Agradeço aos meus co-autores Maria João, Eduardo Curcio e Pedro Amorim. Agradeço aos membros da banca final e da banca de qualificação pela avaliação, comentários e sugestões: Eduardo Uchoa, Rafael Martinelli, Kelly Poldi, Reinaldo Morabito Neto, Luis Meira e Eduardo Xavier. Agradeço à Carla, ao Maycon, ao Rafael Schouery e ao Fábio Hernandes por todas as informações e suporte para ingresso no doutorado. Agradeço aos colegas do IC, do LOCO, da sala 92, de disciplinas e de Waterloo pelo companheirismo e amizade: Yulle, Matheus, Hugo, Phablo, Francisco Jhonatas, Murilo, Marcelo, Rafael Cano, Maurício Zambon, Pedro Hokama, Natanael, Celso, Alan, Ulysses, Renato, Acauan, Atílio, André, Gustavo, Lucas Porto, Vinícius Loti, Klairton, Mário, Rafael Arakaki, Allan, Deyvison, Nishad, Evandro, Alonso, Mônica, Ícaro, Carlos, Francisco Nardi, Camila Tenório, Luã, Rodrigo, Vanessa, Judy, Takeo, Carrilho, Pedro Libório, Roberto, Valderlei, Daniel D. de Oliveira Neto e Priscila. Agradeço aos professores e funcionários do IC: Guilherme P. Telles, André Santanchè, Tomasz Kowaltowski, Lehilton, Cid, Cristiane, Paulo L. de Geus, Pedro J. Rezende e Wilson. Agradeço à equipe do VRPSolver, em especial ao Ruslan Sadykov. Agradeço ainda a: Tony, Mauro Miazaki, Sandro, Ana Elisa, Josiel, Angelita, Ademir, Maurício Figueiredo, Ronaldo, Itana, Letícia, Anderson, Marco Aurélio, Daniel Kikuti e Camila Marchesini. Agradeço o apoio das pessoas dos departamentos/institutos de ensino e pesquisa que fiz/faço parte durante minha vida acadêmica: IC/UNICAMP; Combinatorics and Optimization Department (C&O) da University of Waterloo; Departamento de Ciência da Computação (DECOMP) da Universidade Estadual do Centro-Oeste (UNICENTRO), em Guarapuava; e Departamento de Informática (DIN) da Universidade Estadual de Maringá (UEM). Também agradeço à todas as outras muitas pessoas que fizeram parte desta jornada de doutorado!

Agradeço ao DECOMP/UNICENTRO pelo apoio e pela concessão de afastamento de atividades para cursar o doutorado. Agradeço ao IC/UNICAMP e ao C&O/University of Waterloo pelo apoio e pelos recursos computacionais. Agradeço o suporte financeiro de: Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná (FA); Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – código de financiamento 001; e Natural Sciences and Engineering Research Council of Canada (NSERC).

# Resumo

A ampla categoria de Problemas de Roteamento de Veículos (VRPs) originou-se do VRP Capacitado (CVRP) proposto no fim dos anos 1950, que consiste em encontrar a maneira mais econômica de servir a um conjunto de clientes com uma frota de veículos de uma dada capacidade. Nesta tese, são abordados problemas de otimização combinatória em dois grupos de VRPs: um considerando consumo de energia e outro que considera backhauls. Além de sua relação com custos de transporte, o interesse no consumo de energia em roteamento de veículos está crescendo nas últimas décadas devido ao seu potencial de mitigar impactos ambientais.

Enquanto custos no CVRP são mensurados em termos de distância, o VRP Cumulativo (CmVRP) é uma variante que objetiva minimizar uma medida simplificada de consumo de energia dependente de carga. Combinando os conceitos de duas formulações pré-existentes, este trabalho propõe uma nova formulação de programação linear inteira para o CmVRP, cuja relaxação de programação linear (LP) é mais forte que ambas as relaxações de LP das originais. Além disso, visando resolver a versão inteira do problema, é empregado um algoritmo branch-cut-and-price (BCP) baseado em uma formulação da literatura a fim de apresentar resultados em estado-da-arte para o CmVRP. Adicionalmente, um pequeno resultado de aproximação sobre o problema é apresentado.

Também lidando com um outro aspecto do consumo de energia, este trabalho propõe variantes do CVRP e CmVRP que consideram limites de energia dependente de carga por veículo, o que pode restringir o alcance de cada veículo sem reabastecer ou recarregar energia. Para estes problemas, este trabalho propõe formulações mestre baseadas em partição de conjunto e subproblemas de pricing correspondentes, onde uma das formulações e os subproblemas fazem uso de cargas discretizadas por arco. Deste modo, esta tese apresenta abordagens BCP baseadas nestes componentes. No melhor do nosso conhecimento, esta é a primeira vez que um consumo de energia dependente de carga dos veículos é limitado por meio do subproblema de pricing. São mostrados vastos resultados computacionais comparando o CVRP, CmVRP e suas variantes com energia limitada.

Por fim, este trabalho aborda a variante do CVRP denominada VRP com Backhauls Seletivos, no qual, depois de visitar clientes, cada veículo pode visitar um único backhaul para coletar um valor de receita. Nesta abordagem, considera-se que as receitas são incertas. Para lidar com esta incerteza, um modelo de otimização robusta, para este problema, é elaborado e analisado. São resolvidos os modelos determinístico e robusto por um algoritmo branch-and-cut e são apresentados resultados computacionais comparando-os com uma abordagem heurística.

# Abstract

The large category of Vehicle Routing Problems (VRPs) was originated from the Capacitated VRP (CVRP) proposed in the end of 1950s, which consists in finding the cheapest way to serve a set of customers with a fleet of vehicles of a given capacity. In this thesis, we approach combinatorial optimization problems in two groups of VRPs: one considering energy consumption and other that considers backhauls. Besides its relation to transportation costs, the interest in the energy consumption of vehicle routing is rising in the last decades due to its potential of mitigating environmental impacts.

While costs in the CVRP are measured in terms of distance, the Cumulative VRP (CmVRP) is a variant that aims to minimize a simplified measure of a load-dependent energy consumption. Combining the concepts of two previous formulations, we propose a new integer linear programming formulation for the CmVRP, whose linear programming (LP) relaxation is stronger than both LP relaxations of the original ones. Furthermore, aiming at solving the integer version of the problem, we employ a branch-cut-and-price (BCP) algorithm based on a formulation from the literature to present state-of-the-art results for the CmVRP. Additionally, a small approximation result on the problem is presented.

Also dealing with another aspect of energy consumption, we propose CVRP and CmVRP variants that consider load-dependent energy limits per vehicle, which might restricts each vehicle range without replenishing. To these problems, we propose master formulations based on set-partitioning and correspondent pricing subproblems, where one of the formulations and the subproblems rely on discretized loads per arc. Thereby we devise BCP approaches based on these components. To the best of our knowledge, this is the first time a load-dependent energy consumption of the vehicles is limited by the means of the pricing subproblem. We show extensive computational results comparing the CVRP, CmVRP, and their variants with limited energy.

Finally, we tackle the CVRP variant called VRP with Selective Backhauls, in which, after visiting customers, each vehicle may visit a single backhaul to collect an amount of revenue. In this approach, we consider that the revenues are uncertain. To deal with this uncertainty, a robust optimization model for this problem was devised and analyzed. We solve the deterministic and robust models by a branch-and-cut algorithm and present computational results comparing them with a heuristic approach.

# Contents


**1 Introduction** **11**
- 1.1 Preliminaries . . . 11
- 1.2 Thesis Synopsis . . . 13
- 1.3 Thesis Organization . . . 15

**2 The Arc-Item-Load and Related Formulations for the Cumulative Vehicle Routing Problem** **18**
- 2.1 Introduction . . . 19
- 2.2 Literature Review . . . 21
- 2.3 Fundamentals . . . 22
  - 2.3.1 The Arc-Load Formulation . . . 22
  - 2.3.2 The Set Partitioning $q$-Route Formulation . . . 24
  - 2.3.3 The Arc-Item Formulation . . . 25
- 2.4 Combining Arc-Item with Arc-Load and Related Formulations . . . 26
  - 2.4.1 Why combine? . . . 26
  - 2.4.2 The Arc-Item Coupled to the Arc-Load Formulation . . . 28
  - 2.4.3 The Arc-Item Coupled to the Set Partitioning $q$-Route Formulation 29
  - 2.4.4 The Arc-Item-Load Coupled to the Set Partitioning $q$-Route Formulation . . . 31
  - 2.4.5 The Arc-Item Coupled to the Set Partitioning $t$-Route Formulation 33
  - 2.4.6 Overview of the Formulations . . . 34
- 2.5 Computational Experiments . . . 35
  - 2.5.1 Instances . . . 35
  - 2.5.2 Computational Environment . . . 36
  - 2.5.3 Experiments on Small Instances . . . 36
  - 2.5.4 Experiments on Regular Instances . . . 39
- 2.6 Concluding Remarks . . . 41

**3 Branch-Cut-and-Price Approaches for Vehicle Routing Problems with Load-Dependent Energy Consumption** **43**
- 3.1 Introduction . . . 44
- 3.2 Main Definitions and the Arc-Load Formulation . . . 46
  - 3.2.1 Problems Considered . . . 46
  - 3.2.2 Revisiting the Arc-Load Formulation . . . 47
- 3.3 Review of a Generic BCP Method for VRPs . . . 49
  - 3.3.1 Resource Constrained Paths . . . 50
  - 3.3.2 Master Problem . . . 51
  - 3.3.3 Packing Sets and Elementarity Sets . . . 53

3.3.4 On Pricing Subproblem, Column Generation, and Path Enumeration 53
3.3.5 On Cut Separation . . . 55
3.3.6 On Primal Heuristic, Strong Branching, and Tree Search Strategy . 56
3.4 Modeling CVRP Variants with Energy Consumption . . . 56
3.4.1 Resource Constrained Paths Based on Arc-Loads . . . 56
3.4.2 Master Problems Based on Arc-Loads or Arcs . . . 60
3.4.3 Packing Sets, Elementarity Sets, and Advanced Features . . . 62
3.4.4 Consolidated Methods for the CmVRP . . . 63
3.4.5 Consolidated Methods for the CmVRP-LE and CVRP-LE . . . 64
3.5 Computational Experiments . . . 65
3.5.1 Implementation, Computational Environment, and Parameters . . . 65
3.5.2 Instances . . . 66
3.5.3 Reducing the Number of Arc-Loads and Providing Cutoffs . . . 68
3.5.4 Results for the CmVRP . . . 69
3.5.5 Results for the CmVRP-LE . . . 75
3.5.6 Results for the CVRP-LE . . . 80
3.6 Concluding Remarks . . . 84

**4 Tighter Analysis of an Approximation for the Cumulative Vehicle Routing Problem 87**
4.1 Introduction and Previous Work . . . 87
4.2 Our Contribution . . . 88

**5 A Robust Optimization Approach for the Vehicle Routing Problem with Selective Backhauls 92**
5.1 Introduction . . . 92
5.2 Problem Description and Formulations . . . 94
5.2.1 Deterministic VRPSB . . . 95
5.2.2 Robust VRPSB . . . 98
5.2.3 Estimates of Probabilistic Bounds . . . 102
5.3 Solution Methods . . . 104
5.3.1 Branch-and-Cut . . . 105
5.3.2 Adaptive Large Neighborhood Search . . . 106
5.4 Computational Tests . . . 107
5.4.1 Problem Instances . . . 108
5.4.2 Evaluation of the VRPSB Under Uncertainty . . . 108
5.4.3 Evaluation of Solution Methods . . . 113
5.5 Conclusions and Future Work . . . 118

**6 Final Concluding Remarks 121**
6.1 Contributions . . . 121
6.2 Future Work . . . 123

**Bibliography 125**

**A Full Experimental Results of the Relaxed Arc-Item, Arc-Load, and Arc-Item-Load Formulations for the CmVRP 141**

B Detailed Experimental Results of the Best BCP Approaches for the CmVRP, CmVRP-LE, and CVRP-LE 143

# Chapter 1

# Introduction

A combinatorial optimization problem consists in finding an optimal solution among a discrete set of feasible solutions, i.e., it searches such a set for a solution that minimizes or maximizes an objective function while respects the problem's constraints [136, 178, 8]. A problem of this type is often described as a mathematical programming formulation, which involves decision variables representing components of the solutions, an objective function, constraints given by inequalities, and constraints over the domains of the variables. When the objective function and the constraints are linear and the domains of (at least some of) the variables are integer, we have a mixed-integer linear programming (MILP) formulation [136, 178, 8]. In the general case, a MILP problem is NP-hard, which means that its exact resolution requires an algorithm of superpolynomial computing time when considering $P \neq NP$, i.e., it is unlikely to exist a polynomial-time algorithm for such a problem [47]. Many applied NP-hard problems can be modeled as MILP formulations, as numerous problems in the large category of Vehicle Routing Problems (VRPs) [170]. This category, by turn, originated from the CVRP proposed by Dantzig and Ramser [50], which consists in finding the cheapest way to serve a set of customers with a fleet of vehicles of a given capacity. This thesis employs exact approaches to tackle some VRPs, including CVRP variations that directly consider energy consumption. In the remainder of this chapter, we mention some techniques we rely on to propose resolution approaches that mitigate the hardness of the problems, and give an overview of the contents of this thesis.

## 1.1 Preliminaries

There are several approaches to tackle NP-hard MILP problems, each one with its own trade-offs involving quality of solution and computing time. When the priority is to obtain guaranteed optimal or near-optimal solutions, one might employ an exact method [8, 44]. Given that the problems have a potentially huge number of feasible solutions, exact approaches that perform implicit enumeration are often applied [44, 131, 178]. A typical example is the branch-and-bound (BB) method, that successively divides the solution space while being careful to avoid – by the usage of bounds on solution values – the exploration of sets of feasible solutions with no potential of optimality. Besides, the divisions in the solution space can be seen as branches in a BB tree. As it may be

impractical to wait for an exact answer, the search may be set to finish within a time limit. When the computing time is the major concern, approximation and heuristic methods may be suitable [131]. An approximation algorithm runs in polynomial time and has a formal proof of the quality of the values of the computed feasible solutions compared to the optimal value [173, 35, 177]. Heuristic algorithms, in turn, do not assure a quality of solution and may even not run in polynomial time, however, they often present fast running times [83]. In the following, we give a brief overview of exact approaches for NP-hard MILP problems and cite several works and textbooks that discuss these techniques.

When a problem's MILP formulation is relaxed by dropping the integrality constraints, we are let with a linear programming (LP) formulation, that can be solved in polynomial time [136]. Considering its geometric ties, it is known that the constraints of an LP model defines a polyhedron, whose points correspond to feasible solutions of the problem [39, 21, 8]. One way to solve an LP formulation is by searching only among the vertices of its polyhedron, which is the idea of the Simplex method, that was proposed by Dantzig in 1951 [51, 44] and has been widely used in practice [39, 21, 8]. Besides, some important known facts relate a MILP formulation to its LP relaxation, as follows. Given a MILP formulation, its associated polyhedron contains a (polyhedron given by the) convex hull of its feasible solutions, called mixed-integer hull. It is known that it suffices to search for an optimal solution of the MILP formulation among the vertices of its mixed-integer hull, however, obtaining such a hull is as hard as solving the MILP model itself [8]. Here is where the MILP's LP relaxation comes in: when a solution that fulfill the integrality requirements (of the MILP) is found on a vertex of the LP relaxation's polyhedron, it is guaranteed that such a solution is also on a vertex of the respective mixed-integer hull.

The combination of BB, LP, and upper bounds (that may be obtained heuristically) is a common method to exactly solve MILP problems [178, 8]; where we are considering minimization problems. Also supported by the relation between a MILP formulation and its mixed-integer hull, important aspects of this combined approach are: the feasible solutions of a MILP formulation are also feasible solutions for its LP relaxation; the value of an optimal solution for a relaxed MILP formulation is a lower bound for the values of the feasible solutions of the mixed-integral model, and particularly, for the mixed-integral optimal solutions; and the solution space represented by a MILP formulation of a BB tree node might be divided with the insertion of constraints over a relaxation of it, that, in each child node, typically impose integrality over the domain of a variable. A key point is that the search prunes BB tree nodes by bound, i.e., those whose lower bounds reveal to be higher than the best upper bound computed so far, as the feasible solutions represented by these nodes cannot offer any improvement. When the lower bound of a BB tree node matches the upper bound, that node is fathomed by optimality (of node). The search finishes with a guaranteed optimal solution for the problem once all the created BB tree nodes are fathomed.

In several useful situations, an LP formulation may be defined by an exponential number of variables, constraints, or both [178, 18, 8, 44]. In these cases, we may rely on LP methods that repeatedly solve versions of the model: they start from a small version of the model and add variables (with their columns) and/or constraints on demand, until it becomes sufficient to compute a guaranteed optimal solution. When adding

variables/columns, we have the column generation method, where a so-called pricing subproblem is used to identify columns that, if included in the formulation, may help to reach an optimal solution. On the other hand, when adding constraints, we have the cutting plane method, in which a so-called separation subproblem checks if there is a constraint that is violated by the current solution, and, thereby, should be included in the formulation. It is worth noting that different classes of formulations request specific pricing and/or separation subproblems. Under certain assumptions, this methods run in polynomial or pseudo-polynomial time. For MILP problems, in general, a branch-and-price (BP) method combines the BB with the column generation method to the resolution of the underlying LP formulations [17, 59, 178, 8], while a branch-and-cut (BC) method is similar, but uses the cutting plane method to solve the LP formulations [178, 8]. A branch-cut-and-price (BCP) method, in turn, is a combination of both, that also may have to deal with the extra complexity introduced by the interaction between the new columns and constraints [147, 60, 178]. BCP methods are, for instance, the foundation of the state-of-the-art exact approaches for the classical CVRP and several of its variants [143].

When solving the models of some real-world problems, it might be hard or impossible to have access to the instance data in an exact manner, which may be modeled as uncertainty in the parameters of the formulations [91]. Several approaches of optimization under uncertainty are available in the literature [165, 91], as stochastic programming [27], chance-constrained optimization [91, 119], and robust optimization (RO) [165, 89, 25, 24, 20]. We focus on the RO approach, that aims at finding a solution that is feasible for all possible realizations of the uncertainty data [165, 89, 25, 24, 20]. To avoid a perhaps overly pessimistic approach, the uncertain parameters can be modeled as an uncertainty set for which is possible to adjust the budget of uncertainty [23]. This may allow one to regulate, for instance, a compromise between the conservativeness of the solution and the eventual extra cost that should be paid for that. It is worth adding that, intuitively, the more conservative a solution is, the lower is the probability that it may violate constraints that dependent on uncertain data when the solution is actually applied. A way to tackle a problem in this line is given in the following: first, formulate a deterministic model; then, directly embed the robust aspects; and, after that, using RO techniques, reformulate the model (that typically has an infinity number of constraints) into a linear (or MILP) model with a finite number of constraints, obtaining a robust counterpart of the model [165, 89, 25, 23]. Such a reformulation normally involves the dualization of a subproblem involved with uncertain parameters and the budget of uncertainty [165, 89, 25, 23]. A strong aspect of this approach is that, normally, the robust counterpart formulation is a reasonable MILP model, and thus, may be tackled by the MILP methods presented above [165].

## 1.2 Thesis Synopsis

In this thesis, we approach problems in two groups of VRPs: one considering energy consumption and other that considers backhauls (and uncertainty). Let us first focus on the former group. Besides its relation to transportation costs (e.g., fuel, electric-

ity), the interest in the energy consumption of vehicle routing has risen in the last decades, due to its potential of mitigating environmental impacts (as the amount of greenhouse gas emissions [19, 54]), and is now an established concern [108, 19, 9]. A particular class of problems that consider environment-related aspects is given by the Green VRPs (GVRPs) [55, 9, 111].

One way to take energy consumption into account is by considering it in the objective function. While costs in the CVRP are measured in terms of distance, the Cumulative VRP (CmVRP) is a variant that aims to minimize a simplified measure of a load-dependent energy consumption [108]. In this approach, each arc's energy consumption is defined as the product of the arc distance by the accumulated weight of the vehicle's curb weight and all the items picked since the beginning of the route. We explore integer linear programming (ILP) formulations for the CmVRP, where we use the concepts of two previous formulations to propose a new one, whose LP relaxation is stronger than both LP relaxations of the original ones. Furthermore, we solve the integer version of the CmVRP with a BCP algorithm based on a formulation from the literature, for which we use the generic BCP framework proposed by Pessoa et al. [143], the so-called VRPSolver [156]. The results of our approach improve upon the state-of-art results of Fukasawa et al. [74]. Additionally, we present a small approximation result on the problem (version in which the number of vehicles is not fixed) that represents an improvement over Gaur et al. [80].

Another way to deal with energy consumption is by limiting the energy consumption of each vehicle/route. This is particularly relevant when vehicles need to be restricted by a certain range since recharging or refueling may not be easily done, which is the case of electrical vehicles [45]. In this subject, we propose the CVRP with Limited Energy (CVRP-LE) and CmVRP with Limited Energy (CmVRP-LE), which are, respectively, similar to the CVRP and CmVRP with the difference that the new ones have limited energy per vehicle. We observe that the CmVRP-LE can be seen as a special case of other problem in the literature [86, 185].

Other works consider BP or BCP methods to approach GVRPs and take into account energy consumption, energy limits, and load capacity [58, 99, 30, 36, 179], however, the energy consumption in these works do not depend on the accumulated load. We devise, with the VRPSolver, BCP approaches to the CmVRP-LE and CVRP-LE, that are modeled by a set-partitioning-based master problem formulation and a pricing subproblem [48], where the latter is solved as a Resource Constrained Shortest Path Problem (RCSPP) [151]. Our master problems are formulated with either arc-load or arc variables, while the subproblems rely on arc-loads. An arc-load, by its turn, denotes how much load is carried through an arc [74], and its concept is pivotal to track and limit the load-dependent energy consumption in our modeling of the pricing subproblems. BCP methods with arc-load based master and pricing problems were used before for the CVRP [141] and CmVRP [74], however, to the best of our knowledge, this is the first time a load-dependent energy consumption is limited by the pricing subproblem. This thesis presents extensive computational results for the CmVRP-LE and CVRP-LE with the models we proposed, comparing them with the variants without energy limit.

For the second group of problems that this thesis approaches, consider the VRP with Backhauls (VRPB), that has two different types of customer: the linehaul customers,

which receive goods from a depot (outbound logistics), and the backhaul customers, which send goods back to the depot (inbound logistics) [170]. In the line of the VRPB, we study the VRP with Selective Backhauls (VRPSB) [12], which aims to minimize the routing costs minus the revenues collected at backhaul customers optionally selected. In the VRPSB, selecting one backhaul rather than another depends highly on the revenue it provides, whereas these backhaul revenues may be related to the quality attributes of the products to be collected. However, it might be hard to exactly know the quality attributes beforehand, as is the case of moisture of raw-materials in the wood-base supply chain, that can only be roughly estimated [184, 5, 6]. Notwithstanding the practical application of the VRPSB (e.g., parcel services), few works on it have been described in the literature [92], and, in addition, a recent literature review on the VRPB research shows that no work has yet investigated this problem under uncertainty [109].

In this thesis, we explore the VRPSB under uncertain revenues, where the solutions are required to achieve a minimum revenue from the raw-materials collected at the backhauls. A deterministic version of the problem is given by a MILP formulation and a RO counterpart model is derived. In this robust approach, the revenue uncertainty is modeled as an uncertainty set, where it is possible to adjust the budget of uncertainty to be considered [23]. We devise a BC algorithm to solve the deterministic and robust formulations for the problem, and we also compare them with a heuristic approach. The RO approach shows high potential to tackle the problem, requiring similar computing effort and maintaining the same tractability as the deterministic modeling.

## 1.3 Thesis Organization

Chapters 2, 3, 4, and 5 of this thesis are based on papers we produced for scientific international indexed journals or conferences. Two of these papers are published in international journals, one is being prepared for journal submission, and the other is a short paper published in a conference. General concluding remarks are presented afterwards, in Chapter 6. In the following, we provide descriptions of the papers related to the chapters.

We produced the paper entitled *"The Arc-Item-Load and Related Formulations for the Cumulative Vehicle Routing Problem"*, which was published in the *Discrete Optimization* journal [133]. This paper – with very minor editions – is presented as Chapter 2, and its purpose is to propose several different formulations for the CmVRP and to study their LP relaxations. In particular, the goal is to study formulations based on combining an arc-item concept (that keeps track of whether a given customer has already been visited when traversing a specific arc) with another formulation from the recent literature, the Arc-Load formulation (that determines how much load goes through an arc). Both formulations have been studied independently before – the Arc-Item is very similar to a multi-commodity-flow formulation in Letchford and Salazar-González [116] and the Arc-Load formulation has been studied in Fukasawa et al. [74] – and their LP relaxations are incomparable. Nonetheless, we show that a formulation combining the two (called Arc-Item-Load) may lead to a significantly stronger LP relaxation, thereby indicating that the two formulations capture complementary aspects of the problem. In addition, we

study how set-partitioning-based formulations can be combined with these formulations. We present computational experiments on several well-known benchmark instances that highlight the advantages and drawbacks of the LP relaxation of each formulation and point to potential avenues of future research.

Chapter 3 comprises an extended version of the paper – which is currently in preparation – entitled "*Branch-Cut-and-Price Approaches for Vehicle Routing Problems with Load-Dependent Energy Consumption*". In this paper, we employ a BCP approach to tackle the CmVRP, CmVRP-LE, and CVRP-LE, where, with the exception of the first one, a load-dependent energy consumption is limited by the pricing subproblem. Particularly, we propose the variants of the CVRP and CmVRP with limited energy and devise set-partitioning-based master formulations and correspondent pricing subproblems to solve these problems and the CmVRP, where some formulations and the subproblems rely on discretized loads per arc. We present experiments for all the problems, while we also compare solutions of the CmVRP, CVRP, and their versions with limited energy. The proposed approach can solve instances with up to 101 vertices for the CmVRP, CmVRP-LE, and CVRP-LE.

Incorporated in Chapter 4, we include – with very minor editions – the short paper entitled *"Tighter Analysis of an Approximation for the Cumulative VRP"*, which was published in the proceedings of the *II Encontro de Teoria da Computação* [132]. In this short paper, we deal with an approximation algorithm for the version of the CmVRP that does not fix the number of vehicles/routes in advance. Gaur et al. [80] gave a 4-approximation based on a well-known partition heuristic to the Traveling Salesman Problem (TSP). For this algorithm, we present a tighter analysis, obtaining a $\left(4 - \frac{4}{3s^3Q^2}\right)$-approximation, where $Q$ is the capacity of the vehicle and $s$ is a scaling factor.

A paper entitled *"A Robust Optimization Approach for the Vehicle Routing Problem with Selective Backhauls"* was published in the journal *Transportation Research Part E: Logistics and Transportation Review* [159] as a result of a joint work with Maria João Santos, Eduardo Curcio, and Pedro Amorim from the Faculty of Engineering of the University of Porto, Portugal. Our main contribution to this paper regards the formulation and the exact solving method applied to the problem. In this way, Chapter 5 presents an edited version of this paper that focuses on this topics. In this paper, we explore the VRPSB under uncertain revenues, where a deterministic version of the problem is given by a MILP formulation and a RO counterpart model is derived. We devise a BC algorithm to solve the formulations and compare them with a heuristic approach. Besides, we estimate probabilistic bounds of constraint violation for this formulation.

Finally, we point out aspects related to this thesis organization, i.e., where chapters are based on papers. First, the abstracts of the papers are present in the beginning of chapters. Despite that the contents of these abstracts are mainly presented in this thesis' Abstract and in the current chapter, we believe that they may be of help when reading a chapter in a standalone manner. Secondly, the notation might not be consistent from one chapter to another (although there are not many inconsistencies). For instance, the decision variables that represent the arcs of a route are denoted by different identifiers in chapters 2 and 3. Besides, the definitions of the acronyms are restarted before each of the chapters 2, 3, 4, and 5. Moreover, observe that the word "robust" is related to a property

of cuts in a BCP context, as in Chapter 3, while it refers to an optimization technique in Chapter 5 (and in the current chapter).

# Chapter 2

# The Arc-Item-Load and Related Formulations for the Cumulative Vehicle Routing Problem

**Abstract.** The Capacitated Vehicle Routing Problem (CVRP) consists of finding the cheapest way to serve a set of customers with a fleet of vehicles of a given capacity. While serving a particular customer, each vehicle picks up its demand and carries its weight throughout the rest of its route. While costs in the classical CVRP are measured in terms of a given arc distance, the Cumulative Vehicle Routing Problem (CmVRP) is a variant of the problem that aims to minimize total energy consumption. Each arc's energy consumption is defined as the product of the arc distance by the weight accumulated since the beginning of the route.

The purpose of this work is to propose several different formulations for the CmVRP and to study their Linear Programming (LP) relaxations. In particular, the goal is to study formulations based on combining an arc-item concept (that keeps track of whether a given customer has already been visited when traversing a specific arc) with another formulation from the recent literature, the Arc-Load formulation (that determines how much load goes through an arc).

Both formulations have been studied independently before – the Arc-Item is very similar to a multi-commodity-flow formulation in Letchford and Salazar-González [116] and the Arc-Load formulation has been studied in Fukasawa et al. [74] – and their LP relaxations are incomparable. Nonetheless, we show that a formulation combining the two (called Arc-Item-Load) may lead to a significantly stronger LP relaxation, thereby indicating that the two formulations capture complementary aspects of the problem. In addition, we study how set partitioning based formulations can be combined with these formulations. We present computational experiments on several well-known benchmark instances that highlight the advantages and drawbacks of the LP relaxation of each formulation and point to potential avenues of future research.

## 2.1 Introduction

One of the most studied versions of the class of Vehicle Routing Problems (VRPs) [88, 170] is the classical Capacitated Vehicle Routing Problem (CVRP) [50]. The CVRP can be described as finding a minimum distance set of $K$ routes that start and end at the depot, picking up the customers' demands while respecting the capacities of the vehicles. The Cumulative Vehicle Routing Problem (CmVRP), also known as Energy Minimization Vehicle Routing Problem [107, 108], differs from the former by considering an energy consumption measure as its cost to be minimized. In particular, the cost of traversing an arc now becomes the product of the arc's distance and the load carried while traversing it.

The consideration of the load carried on an arc, in addition to its distance, becomes important as increasing concerns about the environment (and thus the desire to minimize fuel consumption, pollution or related quantities) become more prevalent (see, for example, the surveys Demir et al. [55] and Eglese and Bektaş [66] on Green VRPs). Indeed, Kara et al. [108] mention the use of CmVRP in such a context as a simplified way to minimize the energy consumed. In addition, the energy consumption function of electric unmanned aerial vehicles, also known as drones, [135, 97] and several other types of vehicles are well modeled by the CmVRP [63, 180, 108]. The CmVRP captures critical aspects of multi-package pickup electric drones, as their total weight will increase along the route, affecting the usually restricted autonomy provided by the battery. Thus, the CmVRP is an increasingly important problem to solve in multiple applications.

Several related combinatorial optimization problems have also been studied in the literature. For instance, the CmVRP generalizes the $K$-Traveling Repairman Problem ($K$-TRP) [70, 71, 104], where there are no customer demands and the total cost is given by the sum of the waiting times of the customers. When there is only one repairman, the $K$-TRP is also known as Traveling Repairman Problem (TRP) [2, 7], Traveling Deliveryman Problem [124, 130], Cumulative Traveling Salesman Problem [26], and Minimum Latency Problem [28, 7, 32]. The TRP is also related to the Time Dependent Traveling Salesman Problem (TDTSP) [144, 1, 32]. The version of the CmVRP with a single vehicle was studied by Wang et al. [174], who presented a mathematical model based on Fukasawa et al. [74].

The CmVRP is not only a strongly NP-hard problem, but it has also shown to be quite challenging computationally. The work of Kara et al. [107, 108] initially proposed an exact algorithm for the CmVRP. The best exact results for the problem are from Fukasawa et al. [74], who present both a branch-and-cut (BC) and a branch-and-cut-and-price (BCP) algorithm, where the latter yields the state-of-the-art results. As a matter of fact, the best results for the CVRP also come from a BCP approach [143]. There have also been heuristic approaches for the CmVRP, for instance by Kramer et al. [110] using a matheuristic and by Zachariadis et al. [182] using a local search approach, as well as by Xiao et al. [180]. A variant of the problem with limited duration was addressed by Cinar et al. [40, 41] using constructive algorithms and metaheuristics, whereas the work of Cinar et al. [41] also surveys some methods applied to solve the CmVRP. Heuristic approaches based on linear programming and column generation for the problem were presented by

Gaur and Singh [79] and Singh and Gaur [164]. Without constraining the number of vehicles that can be used, the works of Gaur et al. [80] and Mulati and Miyazawa [132] presented an approximation algorithm for the problem with a factor of less than 4, while Gaur et al. [81, 82] proposed a 6-approximation for the problem with stochastic demands.

While the work of Fukasawa et al. [74] uses arc-load based formulations, whose variables represent if a given weight is carried over a specific arc, the central idea of this work is to combine such a formulation with an arc-item based formulation which keeps track of the individual items that are carried over each arc. We say an arc carries an item $i$ if vertex $i$ has been visited prior to traversing the arc. It is worth noting that the arc-item formulation is very similar to multi-commodity-flow formulations with one commodity per vertex, that have been studied for the CVRP (see for instance Letchford and Salazar-González [116, 117]). Our main contribution in this paper is to combine an Arc-Item formulation for the CmVRP with arc-load ones and to study their theoretical and experimental properties. In particular, we show that one such formulation – the Arc-Item coupled to the Arc-Load, i.e., the Arc-Item-Load (AIL) – strictly dominates other previous formulations of the problem. Following standard notation, we say that a formulation F dominates a formulation G if, for any solution of F's Linear Programming (LP) relaxation, there exists a solution of G's LP relaxation with the same objective function value. Throughout this paper, we also use "relaxation" or "relaxed" to denote the LP relaxation of a formulation. Moreover, in this paper, we adopt the convention that a formulation is said to be "compact" if its number of variables is a polynomial or a pseudo-polynomial function of the size of the problem input data, i.e., the instance.

Since set partitioning based formulations tend to be the most successful in several VRPs, and due to an equivalence between arc-load formulations for the CmVRP and a natural set partitioning formulation, we also consider how the Arc-Item formulation can be combined with such set partitioning formulations. We show one such formulation that achieves the same bound as our compact formulation. While this may indicate that the compact formulation is more useful since no column generation needs to be performed, the set partitioning based formulation can be easily strengthened by using more advanced techniques such as $ng$-routes [14], leading to better bounds. We also propose one natural way to extend the arc-item formulation with a set partitioning one, but show that it may become too hard to solve, since its pricing subproblem is strongly NP-hard. Thus, we propose a final formulation attempting to address this drawback.

We conducted extensive computational experiments comparing the LP relaxations of the formulations of interest over well-known benchmark instances. One particular goal of the research is to gain insight into which formulations are most likely to lead to efficient exact algorithms for the problem.

The other sections of this paper are organized as follows. Section 2.2 presents a literature review of problems and formulations related to our topic. Section 2.3 contains the precise CmVRP definition and other fundamental concepts. Section 2.4 presents our proposed formulations and some of their properties. Experimental results are presented in Section 2.5 and concluding remarks are written in Section 2.6.

## 2.2 Literature Review

Several formulations have been proposed for the CVRP, for instance: two-index and three-index vehicle-flow, one-commodity-flow and multi-commodity-flow, and set partitioning based formulations [103, 162]. Except for the set partitioning based ones, the others are classified as compact formulations. The current best exact algorithms for the CVRP are based on set partitioning formulations combined with families of cutting planes inside a BCP framework [73, 13, 14, 46, 146, 137, 143].

The CmVRP bears many similarities with the CVRP, and most formulations for one of them can be adapted to the other. Indeed, several CVRP formulations have been used in the cumulative problem, for instance one-commodity-flow, arc-load, and set partitioning formulations [108, 74]. Similar to what happens in the CVRP, the best exact approaches to the CmVRP rely on a BCP framework over a set partitioning formulation with additional cutting planes [74]. It is not too surprising that formulations for the CVRP have been adapted to the CmVRP, as the feasible set is the same for both problems, with the only difference arising in how the cost of a solution is calculated. Thus, we devote the rest of this section to reviewing other formulations for the CVRP.

In the series of papers by Letchford and Salazar-González [115, 116, 117], several mixed integer linear programming formulations for the CVRP are presented and/or proposed. We highlight two particular ones which are related to our work, namely MCF1a and MCF2b, based on a multi-commodity-flow idea. Beyond the binary variables $[x_a]$ indicating whether arc $a$ is used, their MCF2b formulation, whose main concept can be traced back to Garvin et al. [78], contains binary variables to indicate whether an arc $a$ is traversed by a vehicle before ($[f_a^i]$ variables) or after ($[g_a^i]$ variables) visiting customer $i$. This kind of formulation also has precedents in modeling the Traveling Salesman Problem (TSP) and some variations [113, 98]. The MCF1a formulation is conceptually similar to the MCF2b, with the difference that only the variables $[x_a]$ and $[f_a^i]$ are present. Moreover, the MCF2b model was strengthened by coupling it to a knapsack based formulation with exponentially many columns, each one corresponding to a packing pattern of the items. The pricing of the variables can be done in pseudo-polynomial time and the resulting formulation is called MCF2K.

Also based on the MCF2b, the Arc-Packing (AP) formulation couples the existing variables with a new set of arc-packing binary variables $[\psi_a^{ST}]$, which take the value of one if and only if a vehicle traverses arc $a$ after having visited the customers (packed) in $S$, and no more, and about to visit the ones (packed) in $T$, and no more. As $S, T \subseteq N$ (where $N$ is the set of all customers), the number of all possible columns is exponential, however its pricing subproblem can be solved in pseudo-polynomial time. In their work, Letchford and Salazar-González [117] were also able to couple the AP to a set partitioning formulation aware of the number of times an arc appears in a non-elementary route, and by this, creating the APnSP. The dominance order is given by APnSP, AP, MCF2K, MCF2b, and MCF1a; where the first dominates the second and so on. With a time limit of two hours (for the first four models), they carried out experiments including relaxations of these formulations over new CVRP instances with 16 vertices, divided into 12 families of 20 instances each, which helped to establish the dominance relationships. It is worth

mentioning that their work deals with the CVRP version that does not fix the number of vehicles in advance.

## 2.3 Fundamentals

We now proceed to formally define the CmVRP, and then move on to describe some basic formulations that will be useful in our later discussions. We also note that we will be working with the *pickup* version of the CmVRP, that is, we are assuming the vehicles are empty to start with and then pick up the customers' demands as they visit them.

**CmVRP.** *The input is given by a complete digraph* $D = (V, A)$ *and a fleet of* $K > 0$ *identical vehicles. The vertices of* $D$ *are given by* $V = \{0\} \cup N$*, where vertex* $0$ *represents the depot, with customers in* $N = \{1, \ldots, n-1\}$*. Each customer* $t \in N$ *has a demand of weight given by a positive integer* $q^t$ *that must be picked up and carried by a vehicle. Every arc* $a \in A$ *has a corresponding distance* $\tilde{d}_a \geq 0$*. Each vehicle has a positive integer capacity* $Q$ *representing the maximum total weight that it can carry and curb weight* $q^0 \geq 0$ *– note that the curb weight does not count towards the capacity. The energy consumed by a vehicle while traversing arc* $a$ *carrying a load* $\ell$ *is* $\tilde{d}_a(q^0 + \ell)$*, and a route performed by a vehicle is defined as a sequence* $\langle 0, v_1, v_2, \ldots, v_l, 0\rangle$ *such that* $\{v_1, v_2, \ldots, v_l\} \subseteq N$ *and* $v_i \neq v_j, \forall i \neq j$*. We define the total demand of such a route to be* $\sum_{i=1}^{l} q^{v_i}$*. The* objective *is to find a set of exactly* $K$ *routes minimizing total energy consumption such that* (i) *they pick up all the customers' demands,* (ii) *each route starts and ends at the depot,* (iii) *each customer is visited exactly once, and* (iv) *the total demand of a route is at most* $Q$*.*

The CmVRP One-Commodity-Flow formulation (CF) was introduced alongside the problem itself by Kara et al. [107, 108]. It has two sets of decision variables over the arcs $a \in A$: binary variables $[x'_a]$ that indicate whether the arcs are traversed by a vehicle, and the continuous $[y'_a] \geq 0$ variables representing the amount of flow passing through the arcs. Conceptually, the flow passing through an arc is equivalent to the load carried on it. The LP relaxation of this formulation is normally solved in a short amount of time. However, the lower bound it provides is weak when compared to the relaxation of the Arc-Load formulation used in Fukasawa et al. [74], and, thus, a branch-and-bound (BB) type algorithm based on the CF formulation tends to explore a larger number of nodes. Thus, we refrain from presenting the CF formulation and present the Arc-Load one next.

### 2.3.1 The Arc-Load Formulation

The CmVRP Arc-Load formulation (AL) was proposed by Fukasawa et al. [74]. It uses decision variables that determine if an arc is being used while carrying a specific weight $\ell$ that is in a discrete set given by $L = \{0, 1, \ldots, Q\}$. The decision variables for the so-called arc-loads are

$$y_a^\ell = \begin{cases} 1 & \text{if the load } \ell \text{ is carried over the arc } a \\ 0 & \text{otherwise} \end{cases}, \qquad \forall a \in A, \ell \in L.$$

We note that such a formulation has been used in other similar problems. Indeed it can be seen as a generalization of a formulation of Picard and Queyranne [144] for the TDTSP.

The formulation is presented below.

$$\text{(AL)} \quad \min \quad \sum_{a\in A}\sum_{\ell\in L} \tilde{d}_a \left(q^0+\ell\right) y_a^\ell \tag{2.1a}$$

$$\text{s.t.} \quad \sum_{a\in\delta^-(0)}\sum_{\ell\in L} y_a^\ell = K, \tag{2.1b}$$

$$\sum_{a\in\delta^+(0)}\sum_{\ell\in L} y_a^\ell = K, \tag{2.1c}$$

$$\sum_{a\in\delta^+(u)}\sum_{\ell\in L} y_a^\ell = 1, \qquad \forall u \in N, \tag{2.1d}$$

$$\sum_{a\in\delta^-(w)} y_a^\ell = \sum_{a\in\delta^+(w)} y_a^{\ell+q^w}, \quad \forall w \in N, \ell \in \{0,\dots,Q-q^w\}, \tag{2.1e}$$

$$y_{u0}^\ell = 0, \qquad \forall u0 \in \delta^-(0), \ell \in \{0,\dots,q^u-1\}, \tag{2.1f}$$

$$y_a^\ell = 0, \qquad \forall a \in \delta^+(0), \ell \in L\backslash 0, \tag{2.1g}$$

$$y_{uv}^\ell = 0, \qquad \forall uv \in A\backslash\delta(0), \ell \in \{0,\dots,q^u-1, Q-q^v+1,\dots,Q\}, \tag{2.1h}$$

$$0 \le y_a^\ell \le 1, \qquad \forall a \in A, \ell \in L, \tag{2.1i}$$

$$y_a^\ell \in \mathbb{Z}, \qquad \forall a \in A, \ell \in L. \tag{2.1j}$$

Note that in this model and throughout this work we may omit parenthesis when referring to arcs, that is, an arc $(u,v)$ might be simply referred to as $uv$. We write $S\backslash e$ as an abbreviation to denote the set difference operation between a set $S$ and a set with only the element $e$, i.e., $S\backslash\{e\}$. The set of arcs that leave a vertex $w$ is denoted by $\delta^+(w)$ and analogously $\delta^-(w)$ is the set of arcs that enter in $w$. The set of all incident arcs of vertex $w$ is denoted by $\delta(w)$.

Constraints (2.1b) and (2.1c) enforce the in-degree and out-degree of the depot to be equal to the number of vehicles. The out-degree of a customer is restricted to one by (2.1d). The load balance of the customers is managed by (2.1e). Constraints (2.1f) enforce that arcs entering the depot must have at least the load weight of the last visited customer. Constraints (2.1g) enforce that arcs leaving the depot must carry no load. For the arcs that are not incident to the depot, constraints (2.1h) block some low and high loads.

In Theorem 2.1, we retrieve a relation between AL and CF formulations. First, we need to define some notation. The feasible region of the LP relaxation of a given formulation is given by $\mathcal{P}$ subscribed by the name of the formulation, for instance $\mathcal{P}_{\text{AL}}$ denotes the LP relaxation of AL. Similarly, we define $\hat{z}^*$ subscribed by the name of the formulation to be the value of an optimal solution for the respective relaxed model.

**Theorem 2.1** (Fukasawa et al. [74])**.** *The formulation AL dominates CF, that is, for any point $y \in \mathcal{P}_{\text{AL}}$, there exists a point $(x',y') \in \mathcal{P}_{CF}$ with the same objective value; particularly, $\hat{z}^*_{\text{AL}} \ge \hat{z}^*_{CF}$. Moreover, there are instances where this dominance is strict.*

### 2.3.2 The Set Partitioning $q$-Route Formulation

We now turn our attention to a formulation based on $q$-routes [38]. A $q$-route is similar to a route, presented in the CmVRP definition, with the difference that it allows the existence of cycles, that is, replacing the condition $v_i \neq v_j, \forall i \neq j$ with the condition $v_i \neq v_{i+1}, \forall i = 1, \ldots, l-1$. We still enforce that the total demand of a route is at most $Q$ and, in the case of a $q$-route, the demand of each customer is accumulated as many times as it is visited. The set of all $q$-routes is denoted as $\Omega$.

In this way, an integer solution for the CmVRP needs to visit each customer $w \in N$ only once by traversing some routes among all in $\Omega$. This can be modeled as a Set Partitioning Problem constrained to choose exactly $K$ routes. Given the decision variables

$$\lambda_r = \begin{cases} 1 & \text{if the } q\text{-route } r \text{ is traversed} \\ 0 & \text{otherwise} \end{cases}, \qquad \forall r \in \Omega,$$

the CmVRP Set Partitioning $q$-Route formulation ($q$R) is as follows:

$$(q\text{R}) \quad \min \quad \sum_{r \in \Omega} c_r \lambda_r \tag{2.2a}$$

$$\text{s.t.} \quad \sum_{r \in \Omega} \lambda_r = K, \tag{2.2b}$$

$$\sum_{r \in \Omega} h_{wr} \lambda_r = 1, \qquad \forall w \in N, \tag{2.2c}$$

$$0 \leq \lambda_r \leq 1, \qquad \forall r \in \Omega, \tag{2.2d}$$

$$\lambda_r \in \mathbb{Z}, \qquad \forall r \in \Omega. \tag{2.2e}$$

The $h_{wr}$ coefficients represent the number of times a customer $w \in N$ is visited by a $q$-route $r \in \Omega$. The cost of a $q$-route $r$ is given by $c_r$ and the number of $q$-routes that can be traversed is constrained by (2.2b). Constraints (2.2c) impose that all customers must be visited exactly once. This model was introduced for the CVRP in Balinski and Quandt [16] and is used in the CmVRP by Fukasawa et al. [74].

As the number of possible $q$-routes is exponential in the size of the graph, the solution of the LP relaxation of $q$R is done using column generation [59, 17], where variables are added to a master LP as needed by solving a pricing subproblem. The use of $q$-routes instead of routes in the $q$R is precisely justified by this method: the pricing subproblem of $q$-routes is weakly NP-hard, while pricing routes is strongly NP-hard. We also note that much stronger set partitioning formulations exist where additional constraints are imposed in $q$-routes. However, for the sake of simplicity, we restrict ourselves to only set partitioning formulations using $q$-routes, but our results can be easily adapted to these other stronger formulations.

Next, we present Theorem 2.2, that tightly links the AL and the $q$R formulations. Note that if a formulation F dominates a formulation G and vice-versa, we say that F and G are equivalent.

**Theorem 2.2** (Fukasawa et al. [74]). *The formulations AL and qR are equivalent, that is, for any point $y \in \mathcal{P}_{AL}$, there exists a point $\lambda \in \mathcal{P}_{qR}$ with the same objective value and vice-versa; particularly, $\hat{z}^*_{AL} = \hat{z}^*_{qR}$.*

### 2.3.3 The Arc-Item Formulation

We now present the CmVRP Arc-Item formulation (AI) that is based on the concept of arc-item, which keeps track of whether a given item is carried over a specific arc. For every $i \in V$, and every $a \in A$ we say item $i$ is carried over arc $a$ if vertex $i$ was visited in the route containing $a$ prior to traversing it. In this way, the decision variables over the arc-items are denoted by

$$x_a^i = \begin{cases} 1 & \text{if item } i \text{ is carried over arc } a \\ 0 & \text{otherwise} \end{cases}, \qquad \forall a \in A, i \in V.$$

The formulation is:

$$\text{(AI)} \quad \min \sum_{a \in A} \sum_{i \in V} \tilde{d}_a q^i x_a^i \tag{2.3a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(0)} x_a^0 = K, \tag{2.3b}$$

$$\sum_{a \in \delta^-(w)} x_a^0 = 1, \qquad \forall w \in N, \tag{2.3c}$$

$$\sum_{a \in \delta^+(w)} x_a^0 = 1, \qquad \forall w \in N, \tag{2.3d}$$

$$\sum_{a \in \delta^-(t)} x_a^t = 0, \qquad \forall t \in N, \tag{2.3e}$$

$$\sum_{a \in \delta^+(t)} x_a^t = 1, \qquad \forall t \in N, \tag{2.3f}$$

$$\sum_{a \in \delta^-(w)} x_a^i = \sum_{a \in \delta^+(w)} x_a^i, \qquad \forall w \in N,\ i \in N \backslash w, \tag{2.3g}$$

$$x_a^i \leq x_a^0, \qquad \forall a \in A,\ i \in N, \tag{2.3h}$$

$$\sum_{i \in N} q^i x_a^i \leq Q, \qquad \forall a \in \delta^-(0), \tag{2.3i}$$

$$0 \leq x_a^i \leq 1, \qquad \forall a \in A,\ i \in V, \tag{2.3j}$$

$$x_a^i \in \mathbb{Z}, \qquad \forall a \in A,\ i \in V. \tag{2.3k}$$

We note that item 0 is always visited before any arcs in all routes and, therefore, variables $x_a^0$ just merely indicate if an arc $a$ is present in some route or not. The objective function (2.3a) minimizes the total cost of all the selected arc-items and, by considering $q^0$ as the curb weight, the objective function correctly considers the total weight (load plus curb weight) picked up prior to traversing $a$.

The selected arc-items whose items are indexed by 0 represent a full route in the classical sense, which also plays the role of a guide route: if an arc-item with item other than 0 is selected in an arc, then the arc-item with item 0 must also be selected in the same arc. With this in mind, we have that Constraint (2.3b) makes the depot out-degree equal the number of vehicles, while constraints (2.3c)–(2.3d) enforce that each customer has, with respect to item 0, an in-degree and out-degree of one. Note that these constraints play the same role that constraints (2.1b)–(2.1d) do for the AL model.

Constraints (2.3e) prevent each item of entering its customer of origin, and constraints (2.3f) enforce that, in fact, each customer's item leaves its origin. Constraints (2.3g) state that all the items entering a given customer $w$, which are not the item from $w$, must also leave it. The aim of constraints (2.3h) is to tie up the arc-items under the guide arc-item 0, i.e., items can pass in an arc only if a vehicle is passing there. Constraints (2.3i) enforce that the total cumulative demand in each route does not exceed the vehicle capacity.

From our experimental results, we state Theorem 2.3 that shows how AI theoretically compares to other formulations.

**Theorem 2.3.** *The formulation AI does not dominate AL, qR nor CF; conversely, neither of these dominate AI.*

As a final note, we remark that the CmVRP AI formulation is very similar to other formulations to the CVRP found in the literature using variables that have the same purpose, e.g., the MCF1a (Letchford and Salazar-González [116]) and the MCF2b (Letchford and Salazar-González [116, 117]). Beyond the objective function, the key difference is how capacity constraints are enforced, which is done in AI via constraints (2.3i), and the fact that some of the other formulations cited above are strengthened by adding more variables and/or constraints. The reason we chose to use formulation AI in this work, even though it is likely weaker than other similar formulations in the literature, is due to its simplicity and the fact that it suffices to capture aspects that are complementary to the AL formulation, as will be shown in the next section. Indeed, Letchford and Salazar-González [117] already mention that MCF2b is somewhat expensive, so we decided that attempting to use these stronger formulations would not be beneficial.

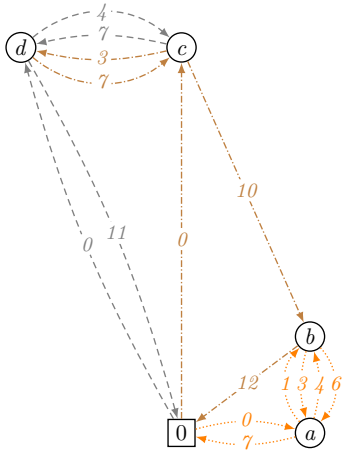
## 2.4 Combining Arc-Item with Arc-Load and Related Formulations

### 2.4.1 Why combine?

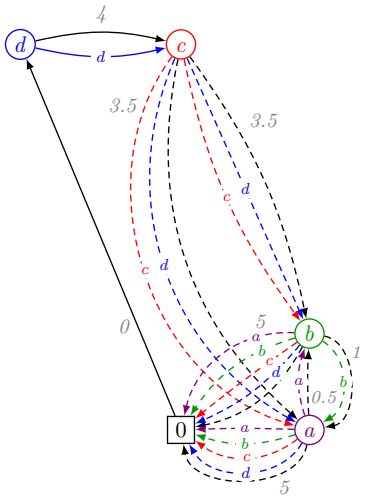
We start by providing an example that illustrates what has been said before, that AI and AL capture complementary aspects of the problem. Intuitively, while the AI formulation captures the precedence relationship (having visited a vertex before going through an arc), the AL captures the load relationship (how much load is carried through an arc). The complementary aspect of these two relationships can be seen by analyzing the literature. For instance, the improvements in some of the multi-commodity-flow formulations that

are similar to AI are made via inequalities that better represent the load carried by a vehicle in an arc or vertex [116]. On the other hand, some of the improvements in the AL formulation [74] come from coupling the arc-load variables with $q$-routes to better capture the precedence relationship.

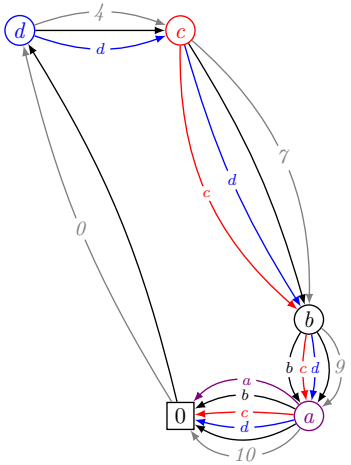
The main idea of formulation AIL (presented in Subsection 2.4.2) is to make sure the load carried in an arc is consistent according to both formulations. Since the load carried in an arc $a \in A$ in the AL formulation is $\sum_{\ell \in L} \ell y_a^\ell$ and in the AI formulation is $\sum_{i \in N} q^i x_a^i$, the formulation makes sure these quantities are the same. In Figure 2.1 we present optimal solutions for a small instance S-n05-k1 ($n = 5$, $K = 1$, and $Q = 15$ with $q^0 = 2.25$) modeled with the AL, AI, and AIL relaxed formulations. We start by looking at the AL solution in Figure 2.1a. Note that as the AL and the $q$R formulations are equivalent (see Theorem 2.2), the solution to AL can be written as a combination of (fractional) $q$-routes. The fractional solution presented in Figure 2.1a can be written as the sum of three fractional $q$-routes, where the decision variables related to the $q$-routes all have value 1/3. Also note that these $q$-routes contain subtours. This allows us to see how combining the AL formulation with the AI formulation may be beneficial, since the AI formulation does not permit visiting a vertex $i$ having already visited it before, which would happen if one follows the $q$-routes.




(a) Formulation AL. The cost is 229.08. All variables have value 1/3. Distinct $q$-routes are drawn in distinct styles.




(b) Formulation AI. The cost is 233.25. The solid arcs are valued 1 and the dashed ones have value 1/2. The letter on the arc represents the item being carried on it.




(c) Formulation AIL. The cost is 235.75 and the solution is integral.

Figure 2.1. Optimal solutions of the LP relaxations of several formulations for instance S-n05-k1. The demands of customers $a, b, c, d$ are, respectively, $1, 2, 3, 4$. The numbers on the arcs represent the load being carried on it.

In Figure 2.1b, we have a half-integral solution for the AI formulation. The solid arc-items have a value of 1 and the dashed ones a value of 1/2. Moreover, the letters over the arc-items represent the item which is being carried along each arc-item. If no letter is shown, item 0 is being carried. For instance, the rightmost depicted arc-item from

customer $c$ to $a$ corresponds to carrying item 0, that is, it corresponds to $x^0_{ca} = 1/2$. The numbers in italic font shape next to the arc-items represent the total load carried from one customer to the other, that is, $\sum_{i \in N} q^i x^i_a$. We will now illustrate why the solution in Figure 2.1b cannot be represented using AL variables. To do this, we will try to write this solution as the sum of $q$-routes, that is, as a solution to $q$R. Note that, by Theorem 2.2, if we show that this solution cannot be written as a solution to $q$R, then it cannot be represented as a solution to AL. One possible way to do so is to use $q$-route $r' = \langle 0, d, c, b, 0 \rangle$ with value $\lambda_{r'} = 1/2$. Note that this would cover all arcs going from $c$ to $b$, since $r'$ carries items 0, $d$ and $c$ along the arc $(c, b)$. However, note that $q$-route $r'$ does not cover $x^a_{b0} = 1/2$, since it does not go through vertex $a$, while the solution on AI covers it. We can see that the only possible other $q$-route that could be used to cover $x^a_{b0}$ combined with $r'$ is $q$-route $r'' = \langle 0, d, c, a, b, 0 \rangle$ with value $\lambda_{r''} = 1/2$. However, that also still leaves $x^b_{ba} = 1/2$ uncovered and there is no other $q$-route that can be used to cover it, when combined with $r'$ and $r''$.

It is not hard to extend the above analysis of Figure 2.1b to show that this solution cannot be represented using AL variables, thus also showing the value of hybridizing the two formulations. Indeed, Figure 2.1c presents an optimal solution of the LP relaxation for the instance S-n05-k1 using the hybridized AIL formulation (presented next, in Subsection 2.4.2), and, as the solution is integer, it is also an optimal solution to the integer version of the formulation. An optimal solution to the hybridized formulation is to use $q$-route $\langle 0, d, c, b, a, 0 \rangle$, which is consistent with both AI and AL formulations.

### 2.4.2 The Arc-Item Coupled to the Arc-Load Formulation

As was previously discussed, the AI formulation is aware of the items carried over each arc via arc-items, and the AL has control over the accumulated load weight in each arc via arc-loads. The idea behind the Arc-Item coupled to the Arc-Load formulation, also referred to as the Arc-Item-Load formulation (or AIL), is to combine those two features. The AIL formulation consists of the entire AI formulation (2.3a)–(2.3k), all the constraints of the AL (2.1b)–(2.1j), and the following additional constraints:

$$x^0_a = \sum_{\ell \in L} y^\ell_a, \qquad \forall a \in A, \tag{2.4a}$$

$$\sum_{i \in N} q^i x^i_a = \sum_{\ell \in L} \ell y^\ell_a, \qquad \forall a \in A. \tag{2.4b}$$

The coupling constraints (2.4a) correspond to the classical CVRP constraints that state that an arc is used (and therefore used carrying item 0) if and only if it is used carrying a load in $L$. As previously mentioned, constraints (2.4b) ensure that the load carried along arc $a$ is consistent between the two formulations.

It is easy to see that constraints (2.1b)–(2.1d) are redundant, and thus can be removed. Thus, the AIL formulation has the variables $[x^i_a]$ and $[y^\ell_a]$, the objective function (2.3a), and the constraints (2.3b)–(2.3k), (2.1e)–(2.1j), and (2.4a)–(2.4b). Furthermore, we have the following theorem about the AIL formulation and its building blocks.

**Theorem 2.4.** *The formulation AIL dominates AI and AL, that is, for any point $(x, y) \in \mathcal{P}_{AIL}$, there exist points $x \in \mathcal{P}_{AI}$ and $y \in \mathcal{P}_{AL}$ with the same objective value; particularly, $\hat{z}^*_{AIL} \geq \hat{z}^*_{AI}$ and $\hat{z}^*_{AIL} \geq \hat{z}^*_{AL}$. Moreover, there are instances where these dominance relations are strict.*

*Proof.* By construction, AIL dominates AI.

Let $(x, y)$ be any solution to the relaxation of the AIL formulation. By construction, we have that $y$ satisfies all the constraints of AL. We now use the coupling constraints (2.4a)–(2.4b) to substitute the variables $[x^i_a]$ by $[y^\ell_a]$ in AIL's objective function (2.3a), obtaining

$$\begin{aligned}\sum_{a\in A}\sum_{i\in V}\tilde{d}_a q^i x^i_a &= \sum_{a\in A}\tilde{d}_a\left(q^0 x^0_a + \sum_{i\in N} q^i x^i_a\right)\\ &= \sum_{a\in A}\tilde{d}_a\left(q^0\sum_{\ell\in L} y^\ell_a + \sum_{\ell\in L}\ell y^\ell_a\right)\\ &= \sum_{a\in A}\sum_{\ell\in L}\tilde{d}_a\left(q^0+\ell\right)y^\ell_a,\end{aligned}$$

which matches the AL's objective function (2.1a), guaranteeing that $z^*_{\text{AIL}} \geq z^*_{\text{AL}}$. Thus, we have that AIL dominates AL.

By Theorem 2.3, AI does not dominate AL and neither does AL dominate AI, completing the proof. □

### 2.4.3 The Arc-Item Coupled to the Set Partitioning $q$-Route Formulation

There is a different alternative to combine the AI formulation with the AL formulation. As mentioned before, Theorem 2.2 says that the $q$R formulation is equivalent to the AL. Therefore, we attempted to combine the $q$R and AI formulations, and this is what this subsection is devoted to.

The new model comprises the entire AI model (2.3a)–(2.3k), all the constraints of the $q$R model (2.2b)–(2.2e), and the coupling constraints

$$x^i_a = \sum_{r\in\Omega}\dot{h}^i_{ar}\lambda_r, \quad \forall a \in A, i \in V, \tag{2.5}$$

where each one of the coefficients given by $\dot{h}^i_{ar}$ represents the number of times an item $i \in V$, picked up at vertex $i$, is carried over an arc $a \in A$ in a $q$-route $r \in \Omega$. In these constraints, we can see that the $[\dot{h}^i_{ar}]$ coefficients, summed through the selected $q$-routes/columns among all the $r \in \Omega$, were precisely tailored to match the $[x^i_a]$ variables.

Using the coupling constraints (2.5), we can simplify the new coupled model. It is not hard to see that Constraint (2.3b) implies Constraint (2.2b) and also that constraints (2.3f) imply constraints (2.2c). Therefore, we can remove the constraints (2.2b)–(2.2c) from the AI$q$R model. The resulting formulation is in the Dantzig-Wolfe explicit master form [147]. The following two results are easy to see.

**Theorem 2.5.** *The formulation AI$q$R dominates $q$R, that is, for any point $x \in \mathcal{P}_{AIqR}$, there exists a point $\lambda \in \mathcal{P}_{qR}$ with the same objective value; particularly, $\hat{z}^*_{AIqR} \geq \hat{z}^*_{qR}$. Moreover, there are instances where this dominance is strict.*

*Proof.* Let $(x, \lambda)$ be any solution to the relaxation of the AI$q$R formulation. By the simplifications we mentioned above, we know that $\lambda$ immediately satisfies the constraints of the $q$R model. Now we use the coupling constraints (2.5) to substitute the $[x^i_a]$ variables in the objective function (2.3a) of AI$q$R, resulting in

$$\sum_{a\in A}\sum_{i\in V} \tilde{d}_a q^i \left(\sum_{r\in\Omega} \dot{h}^i_{ar}\lambda_r\right) = \sum_{r\in\Omega}\left(\sum_{a\in A}\sum_{i\in V} \tilde{d}_a q^i \dot{h}^i_{ar}\right)\lambda_r = \sum_{r\in\Omega} c_r\lambda_r,$$

matching the objective function (2.2a) of the $q$R. Thus, we have that $z^*_{\mathrm{AI}qR} \geq z^*_{qR}$. Our computational experiments show that there are instances where $\hat{z}^*_{\mathrm{AI}qR} > \hat{z}^*_{qR}$, therefore, the main result follows. □

**Theorem 2.6.** *The formulation AI$q$R dominates AI, that is, for any point $x \in \mathcal{P}_{AIqR}$, there exists a point $x \in \mathcal{P}_{AI}$ with the same objective value; particularly, $\hat{z}^*_{AIqR} \geq \hat{z}^*_{AI}$. Moreover, there are instances where this dominance is strict.*

*Proof.* The dominance of the AI$q$R over the AI is guaranteed by the former construction. By Theorem 2.3, the AI does not dominate $q$R, which finishes the proof. □

Despite the above results, there is a major drawback to the AI$q$R formulation: the pricing problem required to solve its LP relaxation via column generation is strongly NP-hard. This is what we focus on next.

Let $\varphi^i_a$, for all $a \in A, i \in V$, be the dual variables associated with the coupling constraints (2.5). The reduced cost $\bar{c}_r$, of a $q$-route/column $r \in \Omega$, can be calculated as

$$\bar{c}_r = c_r - \sum_{a\in A}\sum_{i\in V}\left(-\dot{h}^i_{ar}\right)\varphi^i_a = c_r + \sum_{a\in A}\sum_{i\in V}\varphi^i_a\dot{h}^i_{ar} = \sum_{a\in A}\sum_{i\in V}\varphi^i_a\dot{h}^i_{ar}.$$

The last equation follows since the objective function of the model is based on the $x$ variables, and thus $c_r = 0$.

Thus, the pricing subproblem for the AI$q$R formulation consists of finding $q$-routes $r \in \Omega$ minimizing $\bar{c}_r$ in a way that the total cumulative load of each $q$-route does not exceed the vehicle capacity $Q$. This subproblem is denoted as PRC-AI$q$R. Formally, the PRC-AI$q$R is defined as follows.

**PRC-AI$q$R.** *The input is given by a simple digraph $D = (V, A)$, where $V = \{0\} \cup N$ and vertex 0 represents the depot, with customers in $N = \{1, \ldots, n-1\}$. Each customer $t \in N$ has a demand of weight given by a positive integer $q^t$. For every $a \in A$ and every $i \in V$, there is a cost $\alpha^i_a$ associated with traversing arc $a$ after visiting vertex $i$. The PRC-AI$q$R problem consists then of finding a least cost closed walk $r = \langle 0 = v_0, v_1, \ldots, v_l, v_{l+1} = 0\rangle$ such that $v_j \in N$ for all $j = 1, \ldots, l$; $v_jv_{j+1} \in A$ for all $j = 0, \ldots, l$; and $\sum_{j=1}^{l} q^{v_j} \leq Q$. We note that the cost of such a closed walk must be calculated as follows. For $i \in V$ and $j \in \{0, \ldots, l+1\}$, let $p^i_j = 1$ if $i \in \{v_0, \ldots, v_{j-1}\}$ and 0 otherwise. Then, the cost of $r$ is $\sum_{j=0}^{l}\sum_{i\in V}\alpha^i_{v_jv_{j+1}}p^i_j$.*

We prove in Theorem 2.7 that PRC-AI$q$R is strongly NP-hard by using the directed Hamiltonian cycle problem.

**Theorem 2.7.** *The PRC-AI$q$R is strongly NP-hard.*

*Proof.* We prove the result by showing that PRC-AI$q$R can be used to solve the directed Hamiltonian cycle problem.

Let $D = (V, A)$ be a directed graph. The directed Hamiltonian cycle problem aims to answer the question if $D$ contains or not a directed Hamiltonian cycle. Without loss of generality, we may assume that $V = \{0\} \cup N$ where $N = \{1, \dots, n-1\}$.

The input data for PRC-AI$q$R would, therefore, be the complete digraph $D' = (V, A')$, capacity $Q = n - 1$, and $q^t = 1$ for all $t \in N$. The costs $\alpha_a^i$ would be as follows:

$$\alpha_a^i := \begin{cases} -1 & \text{if } i = 0 \text{ and } a \in A \\ n+1 & \text{if } i \neq 0 \text{ and } v = i \text{ or if } a \notin A\,, \\ 0 & \text{otherwise} \end{cases} \qquad \forall a \in A', i \in V.$$

Consider any solution to PRC-AI$q$R $r = \langle 0 = v_0, v_1, \dots v_l, v_{l+1} = 0 \rangle$. Let $A(r)$ be the set of arcs used in $r$ and $k$ be the number of times the term $\alpha^i_{v_j v_{j+1}} p^i_j$ is equal to $n+1$ in the expression $\sum_{j=0}^{l} \sum_{i \in V} \alpha^i_{v_j v_{j+1}} p^i_j$. The cost of $r$ can be rewritten as $k(n+1) - |A' \cap A(r)|$. Note that due to the capacity constraint, $|A(r)| \leq n$ and so, if $k > 0$, the cost of $r$ is positive. Moreover, note that $k > 0$ if and only if $r$ either uses an arc not in $A$, or visits a customer in $N$ at least two times.

Therefore, the minimum cost solution $r^*$ to PRC-AI$q$R has negative value if and only if $r^*$ is a cycle using only arcs of $A$. Thus, it is easy to see that $D$ has a directed Hamiltonian cycle if and only if $r^*$ has value $-n$. □

We note that the hardness proof relies on the fact that each arc-item combination has a different cost, which in turn comes from the fact that we have one coupling constraint (2.5) for each arc-item combination. Thus, to define a more tractable pricing problem, a more "loosely coupled" formulation becomes necessary. We refrain from defining precisely what loosely coupled means, and proceed to actually presenting the formulation in question.

### 2.4.4 The Arc-Item-Load Coupled to the Set Partitioning $q$-Route Formulation

To construct this new formulation, referred to as AIL$q$R, we start from the entire simplified AIL formulation – given by (2.3a)–(2.3k), (2.1e)–(2.1j), and (2.4a)–(2.4b) – and the $q$R constraints (2.2b)–(2.2e). We couple these together by making use of the following relationship between the arc-loads ($[y_a^\ell]$) and the $q$-routes:

$$y_a^\ell = \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r, \qquad \forall a \in A, \ell \in L, \tag{2.6}$$

where the $\tilde{h}_{ar}^\ell$ coefficient represents the number of times a load $\ell \in L$ is carried over an arc $a \in A$ in a $q$-route $r \in \Omega$. Theorem 2.2 states that the AL and the $q$R formulations are equivalent, and thus we remove the AL constraints (2.1e)–(2.1h).

We now prove that constraints (2.2b)–(2.2c) from $q$R are implied by other constraints that are left in the model, and therefore can also be eliminated. Starting from (2.3b), applying (2.4a), and finally using (2.6), (2.2b) can be derived as follows:

$$\sum_{a\in\delta^+(0)} x^0_a = \sum_{a\in\delta^+(0)}\sum_{\ell\in L} y^\ell_a = \sum_{a\in\delta^+(0)}\sum_{\ell\in L}\sum_{r\in\Omega} \tilde{h}^\ell_{ar}\lambda_r = \sum_{r\in\Omega}\sum_{a\in\delta^+(0)}\sum_{\ell\in L} \tilde{h}^\ell_{ar}\lambda_r = \sum_{r\in\Omega}\lambda_r = K.$$

In the above derivation, we are using the fact that, given a $q$-route $r$ expressed in terms of $\tilde{h}$ coefficients, then, $\sum_{a\in\delta^+(0)}\sum_{\ell\in L}\tilde{h}^\ell_{ar} = 1$.

Similarly, using (2.3d), (2.4a), and (2.6), constraint (2.2c) can be derived, for all $w \in N$, as:

$$\sum_{a\in\delta^+(w)} x^0_a = \sum_{a\in\delta^+(w)}\sum_{\ell\in L} y^\ell_a = \sum_{a\in\delta^+(w)}\sum_{\ell\in L}\sum_{r\in\Omega} \tilde{h}^\ell_{ar}\lambda_r = \sum_{r\in\Omega}\sum_{a\in\delta^+(w)}\sum_{\ell\in L} \tilde{h}^\ell_{ar}\lambda_r = \sum_{r\in\Omega} h_{wr}\lambda_r = 1.$$

This holds since $\sum_{a\in\delta^+(w)}\sum_{\ell\in L}\tilde{h}^\ell_{ar}$ indicates how many times a $q$-route $r$ has visited the vertex $w$, i.e., $h_{wr}$.

In addition, using (2.6), we replace (2.4a)–(2.4b) with the following:

$$x^0_a = \sum_{\ell\in L}\sum_{r\in\Omega} \tilde{h}^\ell_{ar}\lambda_r, \qquad \forall a \in A, \tag{2.7a}$$

$$\sum_{i\in N} q^i x^i_a = \sum_{\ell\in L} \ell \sum_{r\in\Omega} \tilde{h}^\ell_{ar}\lambda_r, \qquad \forall a \in A. \tag{2.7b}$$

Finally, we remove the $[y^\ell_a]$ variables, as well as any other constraints (e.g. nonnegativity) involving them. Note that the model still implicitly uses the concept of arc-load to make the connection between the arc-items and the $q$-routes. Summing up, the AIL$q$R variables are the $[x^i_a]$ and $[\lambda_r]$ and the formulation consists of the objective function (2.3a) and the constraints (2.3b)–(2.3k), (2.2d)–(2.2e), and (2.7a)–(2.7b).

The construction of the AIL$q$R formulation implies the following theorem.

**Theorem 2.8.** *The formulations AIL and AIL$q$R are equivalent, that is, for any point $(x, y) \in \mathcal{P}_{AIL}$, there exists a point $(x, y, \lambda) \in \mathcal{P}_{AILqR}$ with the same objective value and vice-versa; particularly, $\hat{z}^*_{AIL} = \hat{z}^*_{AILqR}$.*

We now show that this new way of coupling the formulations has a more tractable pricing problem. Let $\pi_a$ and $\mu_a$, for all $a \in A$, be the dual variables of the constraints in (2.7a) and (2.7b), respectively. Therefore, the reduced cost $\bar{c}_r$, of a column $r \in \Omega$, can be calculated as given below.

$$\begin{aligned}
\bar{c}_r &= c_r - \left(\sum_{a\in A}\left(-\sum_{\ell\in L}\tilde{h}^\ell_{ar}\right)\pi_a + \sum_{a\in A}\left(-\sum_{\ell\in L}\ell\tilde{h}^r_{ar}\right)\mu_a\right) \\
&= c_r + \sum_{a\in A}\sum_{\ell\in L}\left(\pi_a + \mu_a\ell\right)\tilde{h}^\ell_{ar} \\
&= \sum_{a\in A}\sum_{\ell\in L}\left(\pi_a + \mu_a\ell\right)\tilde{h}^\ell_{ar}.
\end{aligned}$$

The last equation holds since the objective function of the model is based on the $x$ variables, i.e. $c_r = 0$.

Finding the minimum reduced cost $q$-route can be solved as a Shortest Path Problem with Resource Constraints (SPPRC) [102, 151] that minimizes $\bar{c}_r$ while restricting the accumulated load to respect the vehicle capacity $Q$. This version of the problem, referred to as PRC-AIL$q$R, is weakly NP-hard.

### 2.4.5 The Arc-Item Coupled to the Set Partitioning $t$-Route Formulation

We now present a last attempt at combining the AI formulation with a set partitioning based formulation. For this purpose, we introduce the idea of $t$-route, which is a route/path that carries only item $t$. This route/path starts at vertex $t$ and ends at the depot vertex 0, it respects the capacity $Q$, and is allowed to have subtours/subcycles. The main idea is to use $t$-routes to capture the path that an item $t$ will take to reach the depot, once it is picked up. We mimic the guide arc-item concept of AI into the $t$-route of item 0. Formally, let $\Psi_t$ be a set of $t$-routes starting at the vertex $t$. Moreover, let the set of all possible $t$-routes be given by

$$\Omega = \bigcup_{t \in V} \Psi_t.$$

We define $\ddot{h}_{ar}$ as the number of times that an arc $a$ appears in a $t$-route $r \in \Psi_t$ for all $t \in V$. The Set Partitioning $t$-Route formulation ($t$R) we propose is the following.

$$(t\text{R}) \quad \min \sum_{t \in V} \sum_{r \in \Psi_t} c_r \lambda_r \tag{2.8a}$$

$$\text{s.t.} \quad \sum_{r \in \Psi_0} \lambda_r = K, \tag{2.8b}$$

$$\sum_{r \in \Psi_t} \lambda_r = 1, \qquad \forall t \in N, \tag{2.8c}$$

$$\sum_{r \in \Psi_t} \ddot{h}_{ar} \lambda_r \leq \sum_{s \in \Psi_0} \ddot{h}_{as} \lambda_s, \qquad \forall a \in A,\ t \in N, \tag{2.8d}$$

$$0 \leq \lambda_r \leq 1, \qquad \forall r \in \Omega, \tag{2.8e}$$

$$\lambda_r \in \mathbb{Z}, \qquad \forall r \in \Omega. \tag{2.8f}$$

Comparing to the $q$R formulation, the objective function (2.8a) explicitly shows that the number and the role of the $t$-routes differs from those of the $q$-routes. The depot out-degree is now modeled only by the number of $t$-routes that start at it, as stated by the Constraint (2.8b). Constraints (2.8c) ensure exactly one $t$-route is selected for each customer. Constraints (2.8d) ensure that $t$-routes must follow the ones of the depot.

We then propose the AI coupled to the $t$R formulation (AI$t$R). It comprises the entire AI model (2.3a)–(2.3k), all the constraints of the $t$R (2.8b)–(2.8f), and the coupling

constraints

$$x_a^t = \sum_{r \in \Psi_t} \ddot{h}_{ar} \lambda_r, \qquad \forall a \in A, t \in V. \tag{2.9}$$

It is not hard to show that the coupling constraints (2.9) can be used to obtain a simplified version of the model. From constraint (2.3b), the depot out-degree constraint (2.8b) can be obtained. Constraints (2.3f) imply the constraints (2.8c) whereas constraints (2.3h) imply the ones in (2.8d). Furthermore, we present the following theorems.

**Theorem 2.9.** *The formulation AIt$R$ dominates AI, that is, for any point $(x, \lambda) \in \mathcal{P}_{AItR}$, there exists a point $x \in \mathcal{P}_{AI}$ with the same objective value; particularly, $\hat{z}^*_{AItR} \geq \hat{z}^*_{AI}$. Moreover, there are instances where this dominance is strict.*

*Proof.* The result follows by the construction of the formulation AI$t$R and experimental results. □

We can draw the following theorems from our experimental results.

**Theorem 2.10.** *The formulation AIt$R$ does not dominate nor is dominated by AL.*

**Theorem 2.11.** *The formulation AIt$R$ does not dominate AIL.*

We now focus on the pricing problem of AI$t$R. Note that in each column generation round, $n$ pricing subproblems will be run (one for each item). Let $\eta_a^t$, for all $a \in A$, $t \in V$, be the dual variables associated to the coupling constraints (2.9).

Thus, the reduced cost $\bar{c}_r$ of a column $r \in \Psi_t$ can be calculated as

$$\bar{c}_r = c_r - \sum_{a \in A} \left(-\ddot{h}_{ar}\right) \eta_a^t = c_r + \sum_{a \in A} \eta_a^t \ddot{h}_{ar} = \sum_{a \in A} \eta_a^t \ddot{h}_{ar}.$$

Once more, we use the fact that the objective function is based on the $x$ variables, that is, $c_r = 0$.

A pricing subproblem for a specific item $t$ can be treated as a SPPRC that minimizes $\bar{c}_r$ subject to the total accumulated load not exceeding the vehicle capacity $Q$. In each round of column generation, we have to solve this pricing subproblem $n$ times, i.e., one for each item $t \in V$. This version of the problem is called PRC-AI$t$R, and it is weakly NP-hard, as it can be solved by $n$ calls to SPPRC.

### 2.4.6 Overview of the Formulations

We finish this section summarizing the relationship between the formulations of interest in Figure 2.2. We highlight that AL and $q$R are equivalent, and, among the proposed formulations, AIL and AIL$q$R are also equivalent. In this situation, the relations of one automatically apply to the other. We believe that AI$q$R is the strongest formulation in this paper. However, there is no formal proof that it dominates AIL/AIL$q$R and AI$t$R, and moreover its pricing subproblem is strongly NP-hard. The relaxations of the other formulations presented are either compact or have a weakly NP-hard pricing problem.

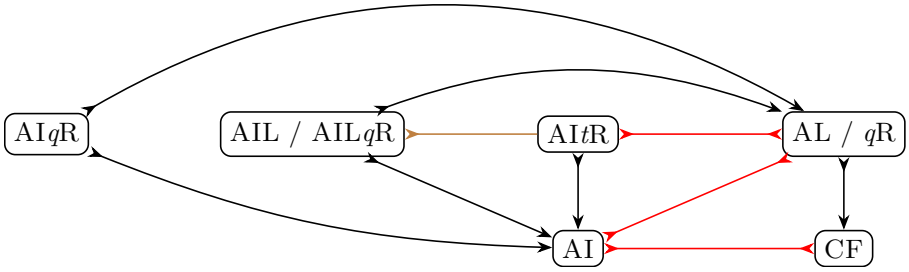



Figure 2.2. Some dominance relations among CmVRP formulations

The pair AIL/AIL$q$R provides a feasible compromise between theoretical quality and computational hardness, as they probably dominate every formulation in Figure 2.2, except possibly AI$q$R. The LP relaxations of either AI and AI$t$R are incomparable with the LP relaxations of the pair AL/$q$R. The AI is incomparable even with the CF. By incomparable, we mean that neither one dominates the other.

## 2.5 Computational Experiments

In this section, we present a set of computational experiments that were made with the intention of gaining a better understanding of the tradeoffs of each model studied in this work.

### 2.5.1 Instances

Some of our experiments were run using a set of benchmark instances that were proposed in previous works. These instances are referred to hereinafter as *regular* instances. The instances of classes A, B, E, and P are obtained from the CVRPLIB[1], while the ones of class V are from the VRDS-COIN-OR[2].

We also propose a new set of *small* instances, which we used to gain some insight into the behavior of the proposed models and also to have a set of instances that could be easily used in the debugging and development phase of our code. This new class of instances for the CmVRP (and CVRP) was named instance class S[3] and it contains eight small instances, namely S-n04-k1, S-n05-k1, S-n08-k2, S-n09-k3, S-n09-k3-d, S-n10-k3, S-n13-k4, and S-n25-k5 where the values of $Q$ are, respectively, 10, 15, 10, 15, 15, 50, 50, and 80.

One final parameter of the CmVRP that does not exist in the CVRP is the curb weight. For our purposes, we defined the vehicle curb weight $q^0$ as the value $\rho Q$, considering $\rho = 0.15$. Note that $q^0$ might not be integer.

[1]CVRPLIB: http://vrp.atd-lab.inf.puc-rio.br
[2]Vehicle Routing Data Sets–COIN-OR:https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm
[3]Instances available online at https://www.loco.ic.unicamp.br/instances

### 2.5.2 Computational Environment

Our code was implemented using the C++ programming language, CPLEX[4] 12.7.1, and C++ Boost graph library[5] (BGL) 1.71. More specifically, we used the SPPRC implementation of the BGL[6]. When dealing with compact formulations, we use CPLEX in the deterministic mode and with just one thread. On the other hand, the column generation formulations are solved with CPLEX in opportunistic mode with four threads. The experiments were run on a computer with four Intel Xeon Gold 6142 @ 2.60 GHz CPU chips and 252 GiB of RAM. The machine has 64 physical cores that can run at most one thread each and the operating system is the GNU/Linux Ubuntu 18.04. The number of test threads we run in parallel is at most the number of cores minus one.

### 2.5.3 Experiments on Small Instances

Table 2.1 presents the results of experiments made on small instances to compare the AL, AI and AIL formulations. Columns "Inst" and "$Q$" show, respectively, the name of the instance and the vehicle capacity $Q$. Column "#" presents the total number of instances. For each of the above formulations, the LP relaxation was solved and the results are presented in columns "$\hat{z}$", "T(s)" and "G(%)", which show, respectively, the optimal value, the time to solve it, and the gap. Note that the gap is calculated as follows: given the best known integer solution value ($z$) and a lower bound value $\hat{z}$, the gap G(%) is given by $((z - \hat{z})/z)100$. In fact, we make use of optimal integer solutions for the entire set of small instances, that were computed using the BCP program of Fukasawa et al. [74]. T(s) has a value of 0 when computations took less than 0.05 s. Finally, column "#I" presents a mark $*$ whenever the solution to the corresponding LP relaxation is integral, and the total number of those in the last row.

Table 2.1. Results of LP relaxations of AI, AL, and AIL on small instances

| | | | Arc-Item | | | | Arc-Load | | | | Arc-Item-Load | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ | # | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) |
| S-n04-k1 | 10 | | 0.0 | 112.50 | * | 0 | 0.0 | 112.50 | * | 0 | 0.0 | 112.50 | * | 0 |
| S-n05-k1 | 15 | | 0.0 | 233.25 | | 1.1 | 0.0 | 229.08 | | 2.8 | 0.0 | 235.75 | * | 0 |
| S-n08-k2 | 10 | | 0.0 | 2,599.00 | | 9.1 | 0.0 | 2,859.00 | * | 0 | 0.0 | 2,859.00 | * | 0 |
| S-n09-k3 | 15 | | 0.0 | 1,078.25 | | 0.2 | 0.0 | 1,074.08 | | 0.6 | 0.0 | 1,080.75 | * | 0 |
| S-n09-k3-d | 15 | | 0.0 | 339.52 | | 8.2 | 0.0 | 360.95 | | 2.4 | 0.0 | 369.75 | * | 0 |
| S-n10-k3 | 50 | | 0.0 | 13,034.45 | | 4.2 | 0.0 | 13,483.67 | | 0.9 | 0.1 | 13,576.25 | | 0.2 |
| S-n13-k4 | 50 | | 0.0 | 908.25 | | 1.7 | 0.0 | 923.50 | * | 0 | 0.3 | 923.50 | * | 0 |
| S-n25-k5 | 80 | | 0.3 | 36,476.00 | | 2 | 0.7 | 37,053.88 | | 0.5 | 84.7 | 37,059.15 | | 0.5 |
| S | | 8 | 0.0 | | 1 | 3.3 | 0.1 | | 3 | 0.9 | 10.6 | | 6 | 0.1 |

The results for the instances S-n05-k1 and S-n08-k2 are enough to imply that AI does not dominate nor is dominated by AL, thus proving part of Theorem 2.3. Moreover, from the numbers in Table 2.1, we see that, in instance class S, AIL has an average gap of

[4] https://www.ibm.com/products/ilog-cplex-optimization-studio

[5] https://www.boost.org/doc/libs/1_71_0/libs/graph/doc/index.html

[6] https://www.boost.org/doc/libs/1_71_0/libs/graph/doc/r_c_shortest_paths.html

0.1%, while AL and AI have average gaps of 0.9% and 3.3%, respectively. Therefore, while AL and AI do not dominate each other, AL performed better than AI. In addition, we point out that, in this instance class, the different structures of AI and AL combined into AIL are able to produce better bounds than each of them individually. However, the major concern is the time spent to solve the LP relaxation of the larger instance S-n25-k5 with AIL.

We also performed experiments using instance class S and the LP relaxation of AIL$q$R. Table 2.2 shows the corresponding results. As this model relies on column generation, the table presents some of that relevant data under the group "CG Stats". The number of column generation rounds is reported in column "# Rnds" and the total number of generated columns is given by "# Gen Cols". The next two columns present the total time spent generating columns ("CG T(s)") and solving the LP master problem ("LP T(s)"). The remaining columns are the same as in Table 2.1. The last row once more presents the total number of instances and average values.

Table 2.2. Results of LP relaxation of AIL$q$R on small instances

| | | | CG Stats | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ | # | # Rnds | # Gen Cols | CG T(s) | LP T(s) | T(s) | $\hat{z}$ | #I | G(%) |
| S-n04-k1 | 10 | | 6 | 30 | 0.0 | 0.0 | 0.0 | 112.50 | * | 0 |
| S-n05-k1 | 15 | | 16 | 128 | 0.0 | 0.0 | 0.0 | 235.75 | * | 0 |
| S-n08-k2 | 10 | | 79 | 169 | 0.0 | 0.5 | 0.5 | 2,859.00 | * | 0 |
| S-n09-k3 | 15 | | 35 | 283 | 0.0 | 0.5 | 0.6 | 1,080.75 | * | 0 |
| S-n09-k3-d | 15 | | 45 | 321 | 0.0 | 0.5 | 0.5 | 369.75 | * | 0 |
| S-n10-k3 | 50 | | 32 | 467 | 0.1 | 0.9 | 1.0 | 13,576.25 | | 0.2 |
| S-n13-k4 | 50 | | 48 | 924 | 0.2 | 2.7 | 2.9 | 923.50 | * | 0 |
| S-n25-k5 | 80 | | 1,473 | 40,614 | 201.2 | 4,223.0 | 4,428.5 | 37,059.15 | | 0.5 |
| S | | 8 | 216.8 | 5,367 | 25.2 | 528.5 | 554.2 | | 6 | 0.1 |

We highlight, from tables 2.1 and 2.2, the expected equivalence of $\hat{z}$ and gap values between AIL$q$R and AIL, as expected due to Theorem 2.8. We highlight the extremely high time to solve the LP relaxation of AIL$q$R for instance S-n25-k5. It takes a total of 4428.5 s, 201.2 s due to the CG steps and 4223 s due to solving the LP master, which means that only 4.5% of the time is used to generate the 40,614 columns through the rounds, and 95.4% of the time is spent solving the 1473 linear programs. In this instance, each linear program takes an average of 2.9 s to be solved. While not much time was spent generating the columns, the high number of rounds implies that a large number of linear programs need to be solved. We note that the total time of solving just the LP relaxation of AIL$q$R for S-n25-k5 (4428.5 s) is extremely high whether compared to solving the LP relaxation of AIL (84.7 s) or even compared to running the exact integer BCP itself (2.9 s).

It is worth pointing out that the LP relaxation of AIL$q$R is solved without enhancements such as cutting planes or column generation improvements such as dual stabilization [65, 142]. Our main goal is to show that the new formulations, AIL and AIL$q$R, provide both in theory and in practice better lower bounds than the previous ones existing in the literature, regardless of the processing time taken. Knowing that and the fact

that their LP relaxations are pseudo-polynomial time solvable, one can implement such improvements to attempt to tighten the bound and decrease the computational cost.

We also performed a computational experiment with the other proposed column generation based formulation, namely AI$t$R. Table 2.3 presents those results on the small instances. When comparing to the average number of rounds of AIL$q$R (216.8), the same statistic in AI$t$R (185.2) is similar, albeit somewhat smaller. However, the number of generated columns is about seven times larger than in the former model, the CG time is about 40 times larger, and the LP time and the total time are about 60 times larger. The main justification of AI$t$R is that it has a pseudo-polynomial time pricing subproblem, such as AIL$q$R. However, even though the number of column generation rounds is almost always smaller, the consistently higher number of generated columns led to larger linear programs. Each linear program took an average of 214.8 s for the S-n25-k5 instance, against the mean of 2.9 s for the AIL$q$R. This gives us some evidence that decreasing the number of rounds may not be good if accompanied by a big increase in the number of generated columns.

Table 2.3. Results of LP relaxation of AI$t$R on small instances

| | | | CG/LP Rounds Specifics | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ | # | # Rnds | # Gen Cols | CG T(s) | LP T(s) | T(s) | $\hat{z}$ | #I | G(%) |
| S-n04-k1 | 10 | | 5 | 94 | 0.0 | 0.0 | 0.0 | 112.50 | | 0 |
| S-n05-k1 | 15 | | 8 | 265 | 0.0 | 0.0 | 0.0 | 233.25 | | 1.1 |
| S-n08-k2 | 10 | | 74 | 788 | 0.1 | 0.1 | 0.2 | 2,657.89 | | 7 |
| S-n09-k3 | 15 | | 24 | 1,260 | 0.1 | 0.2 | 0.3 | 1,078.25 | | 0.2 |
| S-n09-k3-d | 15 | | 36 | 1,194 | 0.1 | 0.2 | 0.3 | 353.40 | | 4.4 |
| S-n10-k3 | 50 | | 24 | 2,591 | 0.3 | 0.9 | 1.3 | 13,282.60 | | 2.3 |
| S-n13-k4 | 50 | | 63 | 5,612 | 1.9 | 5.4 | 7.3 | 910.50 | | 1.4 |
| S-n25-k5 | 80 | | 1,248 | 304,946 | 7,728.1 | 268,072.0 | 275,863.0 | 36,749.63 | | 1.3 |
| S | | 8 | 185.2 | 39,593.8 | 966.3 | 33,509.8 | 34,484.0 | | 0 | 2.2 |

These average numbers are mainly influenced by the test over the instance S-n25-k5. For the sake of completeness, we report some statistics about a test of S-n25-k5 limited to 86,400 s, a more reasonable time limit: it performed 424 rounds generating 296,411 columns, the CG procedure steps took 2174.6 s while the LP solver steps consumed 84,192 s. Note that this suggests that 97.2% of the columns in this test are generated by the first third of column generation iterations.

Another interesting note is that the LP relaxation of AI$t$R for instance S-n04-k1 reaches the same value of $\hat{z} = 112.5$ as the other relaxed models tested, however the solution is fractional. From the results in Table 2.3, one can state that AI$t$R does not dominate the pair AIL$q$R and AIL (implying Theorem 2.11). These results also imply that AI$t$R does not dominate nor is dominated by AL (implying Theorem 2.10), as well as implying that AI does not dominate AI$t$R (proving part of Theorem 2.9).

This paper did not present experimental results over the CF model, as by Theorem 2.1, it is dominated by AL. Moreover, there are also no experiments using AI$q$R due to the fact that solving its relaxation is strongly NP-hard, as stated by Theorem 2.7. In fact, if one is willing to deal with this type of pricing subproblem, a reasonable way is to

directly use a specialized method for the strongly NP-hard Elementary SPPRC [64], and practical experience indicates that several implementation improvements must be made, however that is not the focus of our paper.

### 2.5.4 Experiments on Regular Instances

Experimental results with regular benchmark instances from the literature have their gaps presented in Table 2.4 and their computational times in Table 2.5. These experiments were made over the 100 regular instances, where we set the time limit to 86,400 s (24 h). The reported gaps of the models' LP relaxations were calculated against the objective value of the best available integer solutions.

The integer solutions were obtained as follows. For the majority of the instances, we used the BCP program of Fukasawa et al. [74] with a time limit of 604,800 s (7 days). Besides that, to deal with some instances where some technical issues happened with this BCP program, we relied on our program with the (integer version of the) AIL model and a time limit of 2,592,000 s (30 days). The two instances that ended up with no available integer solution, namely E-n101-k8 and P-n101-k4, are among the largest instances of each class, which leaves us with 98 (out of 100) instances with integer solutions.

To report the main results, first, we kept just the tests carried out over instances that have integer optimal or integer best known solutions available. In addition, we present only those tests where optimal solutions were found for the LP relaxations of all three models, leaving us with 71 instances. We comment that all the LP relaxations of AI and AL were solved within the time limit on all instances. These 71 instances have up to 67 vertices, as can be seen in column $n'$ in Table 2.4, where $n'$ represents the largest instance size in each class that remains after the above mentioned filters were applied. For the non-condensed results correspondent to the results in tables 2.4 and 2.5, see Appendix A.

Table 2.4. Results on gaps of the LP relaxations of AI, AL, and AIL

| | | | Arc-Item | | | | Arc-Load | | | | Arc-Item-Load | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | G(%) | | | | G(%) | | | | G(%) | | | |
| Inst | # | $n'$ | Min | Avg | Max | #I | Min | Avg | Max | #I | Min | Avg | Max | #I |
| A | 19 | 60 | 2.7 | 7 | 12.4 | | 1.1 | 3.6 | 10.1 | | 0.2 | 1.9 | 5.7 | |
| B | 15 | 67 | 3.9 | 9.1 | 22.2 | | 1.3 | 6.7 | 18.9 | | 1.1 | 4.3 | 17.2 | |
| E | 4 | 31 | 5.4 | 10.2 | 19.6 | | 0.7 | 4.3 | 14.1 | | 0 | 3 | 11.4 | 1 |
| P | 20 | 65 | 0.5 | 9.5 | 17.9 | | 0 | 3.2 | 8.5 | 1 | 0 | 1.8 | 5 | 1 |
| V | 13 | 48 | 0 | 9.5 | 22.9 | 1 | 0 | 4.4 | 12.8 | 2 | 0 | 2.5 | 9 | 2 |
| | 71 | 67 | | 8.8 | | 1 | | 4.3 | | 3 | | 2.6 | | 4 |

The average gaps of the relaxations of AI, AL, and AIL, are, respectively, 8.8%, 4.3%, and 2.6%. We can see that the behavior presented in the small instances also happens here, i.e., this sequence of gaps is in decreasing order. Considering we have that AIL dominates AI and AL (see Theorem 2.4), the last value being smaller is expected. However, the consistent ratio between the AIL gap and the others may be attributed

to the fact that each formulation captures different and complementary aspects of the CmVRP, and putting them together tends to produce a stronger formulation that has been worth exploring.

It is worth noting that each instance was preprocessed by having its capacity and demands divided by the greatest common divisor of these values, which we refer to as the scaling factor $\gamma$. The only instances with $\gamma > 1$, are E-n13-k4, E-n22-k4, E-n30-k3, E-n33-k4, and P-n22-k8, whose values of $\gamma$ are, respectively, 100, 100, 25, 10, and 100. The results already take these scaling factors into account, as we multiply the resulting objective function value by $\gamma$. After preprocessing, E-n23-k3, with $Q = 4500$, and E-n33-k4, with $Q = 800$, are the two most significant outlier instances with respect to $Q$; the others have $Q \leq 400$. We mention this as the value of $Q$ is particularly relevant for formulations involving arc-load variables. For instance, for the AL formulation, E-n23-k3 and E-n33-k4 were, respectively, the instances with the first and third highest CPU times (the second highest time was for instance P-n101-k4). For the AIL relaxation, instances E-n23-k3 and E-n33-k4 are the only ones with up to 50 vertices that were not solved under the time limit.

In Figure 2.3, we also present a plot of the cumulative frequency of gaps in the instances. A point $(x, y)$ in this figure represents that $y$ instances have a gap at most $x$%. One can see that the AIL relaxed formulation reaches a gap of at most 5% in 64 instances. For the other two formulations, AL and AI, only 50 and 11 instances have a gap of at most 5%, respectively. Indeed, this chart shows that the average data contained in Table 2.4 are not biased by only a few instances.

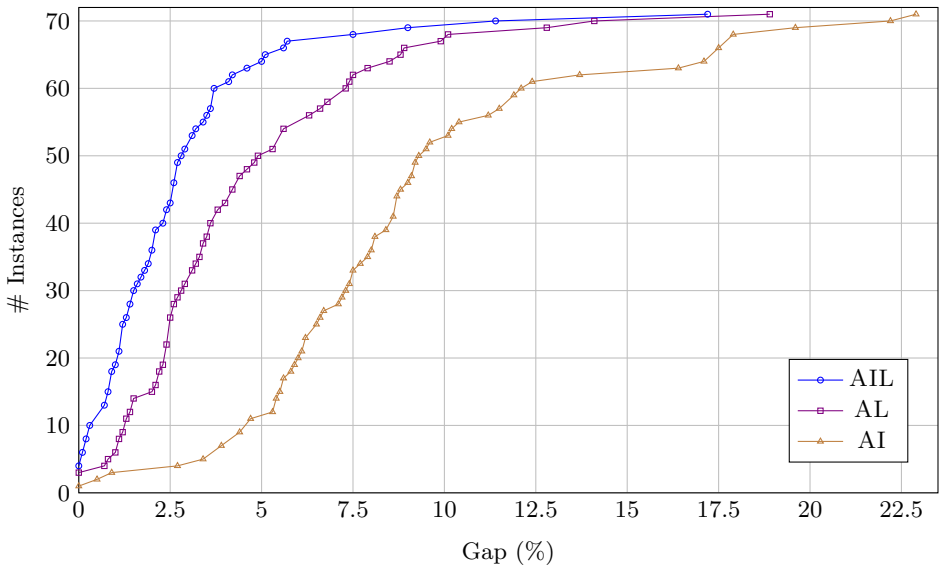



Figure 2.3. Gap cumulative frequency

Despite having presented a promising experimental gap, the main drawback of AIL is the computational time it requires to have its relaxation solved, as shown by Table 2.5. While AI and AL took an average of 16.8 s and 19.6 s in those 71 instances, the AIL computational time average is 17,409.7 s (almost five hours), with some peaks of more than 77,400 s (21.5 h). In addition, we conducted experiments with our current simple

implementation of the AIL$q$R, and, the optimal relaxed solution was reached in only 15 instances within the same time limit of 86,400 s.

Table 2.5. Computational times of the LP relaxations of AI, AL, and AIL

| | T(s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Arc-Item | | | Arc-Load | | | Arc-Item-Load | | |
| Inst | Min | Avg | Max | Min | Avg | Max | Min | Avg | Max |
| A | 1.0 | 14.8 | 83.1 | 3.9 | 17.5 | 39.4 | 709.9 | 16,120.9 | 81,763.9 |
| B | 1.1 | 16.4 | 121.8 | 3.4 | 18.7 | 51.9 | 885.9 | 30,802.6 | 77,429.0 |
| E | 0.0 | 1.3 | 4.1 | 0.0 | 5.4 | 15.5 | 0.4 | 2,544.0 | 9,316.0 |
| P | 0.0 | 11.6 | 68.9 | 0.0 | 37.9 | 143.0 | 0.1 | 22,189.2 | 85,160.6 |
| V | 0.0 | 33.3 | 224.5 | 0.0 | 0.1 | 0.6 | 0.4 | 1,060.9 | 5,043.2 |
| | | 16.8 | | | 19.6 | | | 17,409.7 | |

## 2.6 Concluding Remarks

In this paper, among other advances, we worked on several compact formulations for the CmVRP: we described the AI and the AL formulations from the literature; and proposed a combination of both, the AIL formulation. We then stated that AIL dominates AI and AL and presented dominance relations between all these formulations and the classical CF. Our experiments show a significant gain in the LP relaxation gap of AIL, when compared to AI and AL. The advantage of the AIL may be attributed to the fact that each base formulation alone captures different and complementary aspects of the CmVRP, and putting them together tends to produce a stronger formulation that is worth exploring. The downside is the large amount of time spent in solving the tighter formulation.

We have also proposed the set partitioning AI$t$R and AIL$q$R formulations, that have pricing subproblems solvable in pseudo-polynomial time. While the former did not provide a computational advantage, important results were derived about the latter. We proved that AIL and AIL$q$R are equivalent, and besides, this relation was also important to provide an intuition about the good gap results of AIL when compared to the AI and the AL formulations individually. We also contributed by proposing a somewhat natural way to try and combine AI with a set partitioning formulation, namely the AI$q$R formulation, which dominates the $q$R formulation, and thus, by transitivity, also dominates AL. However, such a natural formulation is not likely to be useful in practice since its pricing subproblem is strongly NP-hard.

The results of this paper point to multiple possibilities of research. The strength of the AIL can be further studied in order to discover its relation with the several families of cuts for the CVRP available in the literature, e.g., proving whether the cuts are useful over the model, including cuts which eliminate cycles of length two, which already exist for the AL. The experimental results concerning the gaps suggests one may benefit from using AIL in BB and BC methods, but further research is needed to speed up its high computational cost. Considering the AIL and AIL$q$R equivalence, speeding up the column generation

process and also strengthening it by using other structures such as *ng*-routes [14] may make the latter viable to be used in a BP or BCP method. It is also worth searching for a formulation whose LP relaxation *(i)* can be solved in pseudo-polynomial time, and *(ii)* gives bounds that are better than the ones from AIL*q*R.

## Acknowledgments

The author M.H. Mulati acknowledges the financial support of Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná (FA), Secretaria de Estado da Ciência, Tecnologia e Ensino Superior do Paraná (SETI), Governo do Estado do Paraná, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). M.H. Mulati had scholarship of / *foi bolsista da* CAPES / Programa de Doutorado Sanduíche no Exterior (PDSE) / Processo nº 88881.190068/2018-01. The author also acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The author R. Fukasawa acknowledges the support of the NSERC [funding reference number RGPIN-2020-04030]. This research is partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) procs. 425340/2016-3 and 314366/2018-0, and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) procs. 2015/11937-9 and 2016/01860-1.

# Chapter 3

# Branch-Cut-and-Price Approaches for Vehicle Routing Problems with Load-Dependent Energy Consumption

**Abstract.** Vehicle Routing Problems (VRPs), in general, aim at finding a least cost set of routes to serve customers with a fleet of vehicles under given constraints. While costs in the classical Capacitated VRP (CVRP) are measured in terms of distance, the energy consumption of vehicle routing is currently an established topic due to its potential of mitigating environmental impacts. The energy consumption might be taken into account in the objective function, as is the case in the variant called Cumulative VRP (CmVRP), or used to limit the energy that a vehicle can consume over a route, thus restricting its range without replenishing. It is important to note that the energy consumption of a vehicle may be significantly affected by the weight it carries, which, in turn, can be significantly affected by the loads picked up when visiting customers.

In this paper, we employ a branch-cut-and-price (BCP) method to tackle CVRP variants that consider a simplified load-dependent energy consumption measure. Branch-and-price and BCP approaches with energy considerations were done before, however, to the best of our knowledge, this is the first time a load-dependent energy consumption is limited by the pricing subproblem. Particularly, we propose variants of the CVRP and CmVRP with limited energy and devise set-partitioning-based master formulations and correspondent pricing subproblems to solve these problems and the CmVRP, where some formulations and the subproblems rely on discretized loads per arc.

Our BCP approaches to the CmVRP reports computational results that improve upon the state-of-the-art ones of Fukasawa et al. [74]. We present experiments comparing solutions of the CmVRP, CVRP, and their versions with limited energy. The proposed approaches can solve instances with up to 101 vertices for the CmVRP and the versions with limited energy of the CmVRP and CVRP.

## 3.1 Introduction

The numerous and still very actively studied category of Vehicle Routing Problems (VRPs) [170] was originated from the Capacitated VRP (CVRP) [50]. A description of the CVRP is that it aims to find a set of vehicle routes, starting and ending at the depot, that picks up the customers' demands and minimizes the traveled distance, where each route respects vehicle load capacity. The problem is defined on top of a digraph $D = (V, A)$, where the depot and customers comprise the vertex set $V$ and each arc in $A$ has an associated distance.

A significant amount of research has been devoted to studying other measures of quality of a route (besides distance). A particular one that has received significant recent attention is energy consumption, given its relationship to transportation costs (e.g., fuel, electricity) and amount of greenhouse gas emissions [19, 54], which is an established environmental concern [108, 19, 9]. One way to take energy consumption into account is by considering it in the objective function. Another one is by imposing a constraint that limits the energy consumption of each route. This latter choice is particularly relevant when vehicles need to be restricted by a certain range since recharging or refuel may not be easily done, which is the case of electrical vehicles [45]. One simplified measure of energy consumption is proposed in Kara et al. [108], who study the so-called Cumulative VRP (CmVRP) in which energy is measured by multiplying the distance traveled when traversing an arc by the accumulated load weight carried over it. We note that, even though it is a simplified measure of energy consumption, it takes already an important consideration into account, namely the load carried. Indeed, Goeke and Schneider [86] pointed out the importance of considering the load in the energy consumption, as it impacts, for instance, the range of the vehicles.

There have been several other works that consider energy in the context of routing. We will now review a few of these works in order to better contextualize the problem we chose to tackle. A particular class of problems, that has considered variants that take into account environment-related issues is the class of Green VRP (GVRP), as surveyed, for instance, in Demir et al. [55], Eglese and Bektaş [66], Pelletier et al. [139], Erdelić and Carić [67], Asghari and Mirzapour Al-e-hashem [9, 10], Kucukoglu et al. [111]. There are different subclasses of GVRPs that differ in how energy is accounted for and/or limited. For instance, the Pollution-Routing Problem (PRP) has a comprehensive objective function that aims at minimizing the cost of fuel, cost of greenhouse gas emissions, and drivers' wage [19]; thereby, the amount of energy consumed over each arc depends on several factors such as distance, vehicle curb weight, current load and speed, and vehicle and road specifications. The Alternative Fuel Green VRP (AFGVRP) assumes the usage of alternative fuel, as, for example, biodiesel, electricity, ethanol, and hydrogen [68], where the fuel is consumed proportionally to the distance traveled and each vehicle has fuel tank capacity. In AFGVRPs, refueling stations are commonly used but, unlike several other types of VRP, no limit is imposed on the total load carried by each vehicle. In a basic Electric VRP (EVRP) (see the survey Kucukoglu et al. [111]), the vehicles consume some amount of the total energy available in their battery proportionally to the distance traveled [45, 161, 58]. The works of Goeke and Schneider [86], Zhang et al. [185] have

considered EVRP variants where the energy consumption takes into account the effect of the distance, curb weight, current load, speed per arc, and vehicle and road specifications, besides also considering drivers' wage. One final comment is that several other factors have been taken into account in the literature. For instance, the presence of recharging stations are common in EVRPs and time windows are common in GVRPs.

One important consideration in the abovementioned works is the methodology used to solve the resulting models. The number of works that consider branch-and-price (BP), branch-cut-and-price (BCP), and tailored branch-and-cut (BC) approaches to GVRPs is rather small, whilst the majority of studies used mixed integer linear programming (MILP) formulations that are simply fed directly to a standard solver, other exact methods, and heuristic methods, as reported in the surveys Asghari and Mirzapour Al-e-hashem [9], Kucukoglu et al. [111]. Given the success of BCP in solving several types of VRPs [170, 143] this is a significant gap in the literature of GVRPs. The only works that consider BP or BCP methods to approach GVRPs and take into account energy consumption, energy limits, and load capacity are Desaulniers et al. [58], Hiermann et al. [99], Breunig et al. [30], Ceselli et al. [36], Wu and Zhang [179]. However, the energy consumption in these works do not depend on the accumulated load, and Desaulniers et al. [58] indeed mention that the usage of the accumulated load in the energy consumption would lead to a more complex pricing subproblem.

To deal with the limited energy capacity of the vehicles, we propose the CVRP with Limited Energy (CVRP-LE) and CmVRP with Limited Energy (CmVRP-LE), which are, respectively, similar to the CVRP and CmVRP with the difference that the new ones have limited energy per vehicle, where we employ the same energy consumption function of the CmVRP. As the CVRP and CmVRP [108], the new problems are strongly NP-hard. In relation to other problems in the literature, Kramer et al. [110] noted that the CmVRP can be seen as a simplified version of the PRP and we observed that the CmVRP-LE can be seen as simplified versions of the EVRPs approached by Goeke and Schneider [86], Zhang et al. [185], whereas, surprisingly, we did not find in the GVRP literature a problem that can accommodate the CVRP-LE as a special case. We chose to use this energy consumption as it captures the main aspect of making energy consumption dependent on the load, while already incorporating a significant challenge in terms of pricing, as mentioned in Desaulniers et al. [58]. We also highlight that the CmVRP and CmVRP-LE may be good choices to model critical aspects of multi-package pickup electric drones [135, 97, 63, 133], as their carried load weight will increase along the route, affecting the usually restricted range provided by the battery. If the drones' traveled distance or time duration is the main concern, the CVRP itself and, especially, the CVRP-LE may be good choices of modeling.

In this paper we devise BCP approaches for the CmVRP, CmVRP-LE, and CVRP-LE, which are based on the generic framework proposed by Pessoa et al. [143], the so-called VRPSolver [156]. Fitting in this BCP framework, each of our problems of interest are given by means of a MILP master problem formulation and a pricing subproblem [48], where the latter is modeled as a Resource Constrained Shortest Path Problem (RCSPP) [151]. Our master problems are formulated with either arc-load or arc (vehicle-flow) variables, which are coupled to resource constrained path variables; we

point out that the master formulations are set-partitioning based. An arc-load, by its turn, denotes how much load is carried through an arc [74], and its concept is pivotal in our modeling of the pricing subproblems: the arc-loads usage allow the pricing procedure to track and limit the load-dependent energy consumption that varies throughout the vehicles' routes. To accomplish this, we define multi-digraphs whose arcs are given by arc-loads to be used as inputs to the subproblems modeled as RCSPPs. BCP methods with arc-load based master and pricing problems were used before for the CVRP [141] and CmVRP [74], however, to the best of our knowledge, this is the first time a load-dependent energy consumption is limited by the pricing subproblem.

In summary, the main contributions of this work are: *(i)* proposing CVRP variants that consider load-dependent energy limits – i.e., the CmVRP-LE and CVRP-LE – and devising BCP approaches to solve them; *(ii)* reporting state-of-the-art exact results for the CmVRP that improve upon the BCP approach of Fukasawa et al. [74]; *(iii)* presenting extensive computational results for the CmVRP-LE and CVRP-LE with the models we proposed, where we compare them with the variants without energy limit; and *(iv)* the usage of a pre-processing algorithm to avoid the creation of arc-loads that represent unpackageable amounts of loads.

This paper is organized as follows. Section 3.2 presents fundamental concepts, models, and the pre-processing algorithm of arc-loads we are using. Section 3.3 revises the aspects of the generic BCP framework used in our work. In Section 3.4 we propose new problems, formulations, and approaches to deal with them. Our experimental computational results are given in Section 3.5. In Section 3.6 we present our concluding remarks and point to future work.

## 3.2 Main Definitions and the Arc-Load Formulation

In this section, we provide formal definitions of the several VRP variants of interest and revisit the arc-load (AL) formulation. We also provide a pre-processing algorithm with the aim to decrease the number of variables needed in the building of this model.

We emphasize that we are dealing with the pickup version of the VRP. We start by introducing some basic notation. Consider a digraph $D = (V, A)$, where $V$ is the set of vertices and $A$ is the set of arcs. An arc $(u, v) \in A$ may also be referred to as $uv$. For a vertex $w \in V$, the set of arcs that leave it is denoted by $\delta^+(w)$ and, analogously, the set of arcs that enter it is written as $\delta^-(w)$. In addition, the set of all arcs that leave from or enter in $w$ is given by $\delta(w) = \delta^-(w) \cup \delta^+(w)$.

### 3.2.1 Problems Considered

Let us formally define the problems CVRP, Symmetric Capacitated VRP (SymCVRP), CmVRP, CVRP-LE, and CmVRP-LE. The following definitions are common for all of these problems. The input is given by a complete digraph $D = (V, A)$ and a fleet of $K > 0$ identical vehicles. The vertices of $D$ are given by $V = \{0\} \cup N$, where vertex 0 represents the depot and the customers are given in $N = \{1, \ldots, n-1\}$. Each customer $w \in N$ has a demand of weight given by a positive integer $q^w$ that must be picked up and carried

to the depot by a vehicle. Every arc $a \in A$ has a corresponding distance $d_a \geq 0$. Each vehicle has a positive integer capacity $Q$ representing the maximum load weight that it can carry. A route performed by a vehicle is defined as a sequence $\langle w_0, w_1, \ldots, w_{l-1}, w_l \rangle$ such that $w_0 = w_l = 0$, $\{w_1, w_2, \ldots, w_{l-1}\} \subseteq N$, and $w_i \neq w_j$ for all $i \neq j$ such that $1 \leq i, j \leq l-1$. We define the total demand of such a route to be $\sum_{i=1}^{l-1} q^{w_i}$. A feasible solution is defined by a set of exactly $K$ routes such that *(i)* they pickup all the customers' demands, *(ii)* each customer is visited exactly once, and *(iii)* the total demand of a route is at most $Q$.

In the **CVRP**, the *objective* is to find a feasible solution that minimizes total distance. The **SymCVRP** can be described as a CVRP where every pair of arcs $uv$ and $vu$ have same distance, which is equivalent to defining the problem on an undirected graph. The other problems also have a vehicle curb weight given by a non-negative integer $q^0$ – note that the curb weight does not count towards the capacity. The energy consumed by a vehicle while traversing an arc $a$ carrying a load $\ell_a$ is $d_a(q^0 + \ell_a)$. In the **CmVRP**, the *objective* is to find a feasible solution that minimizes total energy consumption.

For the two problems with limited energy, consider that each vehicle has a positive integer energy limit $E$. We define the energy consumption of a route to be $\sum_{i=0}^{l-1} d_{w_i w_{i+1}} \left(q^0 + \ell_{w_i w_{i+1}}\right)$. For these two problems, in addition to requiring *(i)*–*(iii)*, it must also be true that *(iv)* the energy consumption of each route is at most $E$. Like their counterparts that have no energy limit, the *objective* in the **CVRP-LE** is to find a feasible solution that minimizes total distance, while the *objective* in the **CmVRP-LE** is to find a feasible solution that minimizes total energy consumption.

### 3.2.2 Revisiting the Arc-Load Formulation

In VRPs studied in this paper, an amount of load is carried over an arc when it is traversed by a vehicle in its route. We consider that these loads are always integral. We use the term *arc-load* to represent an arc being used to carry a specific load. Formally, we define arc-load $(a, l)$ to represent arc $a$ being used to carry load $l$. We use, interchangeably, the following alternative notations for arc-load $(a, l)$: $a^\ell$, $(uv)^\ell$, or $(u, v)^\ell$.

Thus, given the domain set of discrete loads of an arc, there will be one arc-load for each load in that arc. Formulations based on arc-load were used for the CVRP [141, 137] and the CmVRP [74, 133], as well as for similar problems, namely, the CVRP with unitary demands [85] and the Time Dependent Traveling Salesman Problem [144].

Let $\mathcal{L}(uv)$ denote the domain set of the discrete loads that can be associated with each arc $uv \in A$, which is defined by

$$\mathcal{L}(uv) = \begin{cases} \{0\}, & \text{if } uv \in \delta^+(0), \\ \mathcal{L}'\left(uv, [q^u, Q - q^v]\right), & \text{if } uv \notin \delta(0), \\ \mathcal{L}'\left(uv, [q^u, Q]\right), & \text{if } uv \in \delta^-(0), \end{cases}$$

where we can see that only loads valued 0 can go out the depot. The other cases are defined by $\mathcal{L}'\left(uv, [\ell^-, \ell^+]\right)$, that denotes the set of all loads corresponding to the combinations of all customers' demands (always including $u$ and excluding $v$) that are packageable as loads

in the interval $[\ell^-, \ell^+]$. To compute each arc's *discrete load domain as packageable demands* of the customers, we applied a simple algorithm based on sets, that is presented in Algorithm 3.1. The main idea of this algorithm resembles the idea of discretization points using dynamic programming commonly encountered in the literature of packing/cutting problems as, for instance, in Cintra et al. [42], de Queiroz et al. [53], Hokama et al. [101]. Note that the number of loads/combinations on an arc is upper bounded by the vehicle capacity $Q$. Similarly, the algorithm's time and space complexities are upper bounded by a function that is pseudo-polynomial in $Q$. Moreover, given a suitable data structure to represent sets, it is convenient to deal with the elements of set $L$ sorted by non-decreasing load order, as denoted by line 8.

Algorithm 3.1. Computing each arc's discrete load domain as packageable demands

```
COMPUTE-L'(uv, [ℓ⁻, ℓ⁺])
1   L = {ℓ⁻}
2   for w ∈ N \ {u, v} do
3       L' = ∅
4       for ℓ ∈ L do
5           if ℓ + q^w ≤ ℓ⁺ then
6               L' = L' ∪ {ℓ + q^w}
7       L = L ∪ L'
8   SORT(L)
9   return L
```

With this in hand, we state the decision variables to be used to represent the arc-loads, that are:

$$y_a^\ell = \begin{cases} 1 & \text{if the load } \ell \text{ is carried over the arc } a, \\ 0 & \text{otherwise,} \end{cases} \qquad \forall a \in A, \ell \in \mathcal{L}(a). \tag{3.1}$$

The AL formulation is given in the following.

$$\text{(AL)} \quad \min \; f(y) \tag{3.2a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(0)} \sum_{\ell \in \mathcal{L}(a)} y_a^\ell = K, \tag{3.2b}$$

$$\sum_{a \in \delta^+(w)} \sum_{\ell \in \mathcal{L}(a)} y_a^\ell = 1, \qquad \forall w \in N, \tag{3.2c}$$

$$\sum_{\substack{a \in \delta^-(w): \\ \ell \in \mathcal{L}(a)}} y_a^\ell - \sum_{\substack{a \in \delta^+(w): \\ \ell + q^w \in \mathcal{L}(a)}} y_a^{\ell + q^w} = 0, \qquad \forall w \in N, \; \ell \in \mathcal{L}(\delta^-(w)), \tag{3.2d}$$

$$y_a^\ell \geq 0, \qquad \forall a \in A, \; \ell \in \mathcal{L}(a), \tag{3.2e}$$

$$y_a^\ell \in \mathbb{Z}, \qquad \forall a \in A, \; \ell \in \mathcal{L}(a). \tag{3.2f}$$

The function $f(y)$ to be minimized in (3.2a) is chosen according to the problem type. The CVRP aims at minimizing the travel distance. On the other hand, in the CMVRP

case, we use an objective function that minimizes a simplified measure of energy consumption, which is given by the sum of the energy consumed in each selected arc-load. The expressions are given in the following:

$$f(y) = \begin{cases} \sum_{a \in A} \sum_{\ell \in \mathcal{L}(a)} d_a y_a^\ell, & \text{if minimizing distance,} \quad (3.3a) \\ \sum_{a \in A} \sum_{\ell \in \mathcal{L}(a)} d_a \left(q^0 + \ell\right) y_a^\ell, & \text{if minimizing energy consumption.} \quad (3.3b) \end{cases}$$

Constraint (3.2b) makes the out-degree of the depot be equal to the number of vehicles. Note that, for each arc $a \in \delta^+(0)$, only the arc-load with load 0 exists, i.e., $a^0$. Constraints (3.2c) ensure that each customer has an out-degree of 1. Constraints (3.2d) ensure that the load leaving node $w$ is equal to the load entering it plus $q^w$.

We end this subsection by remarking that the main difference between the AL formulation for the CVRP, denoted by (3.3a) and (3.2b)–(3.2f), and the AL formulations for this problem in Pecin et al. [137] and Pessoa et al. [141] is that, on those works, the employed definition of $\mathcal{L}(uv)$, for each $uv \in A$, is denoted by

$$\mathcal{L}(uv) = \begin{cases} \{0\}, & \text{if } uv \in \delta^+(0), \\ \{q^u, \ldots, Q - q^v\}, & \text{if } uv \notin \delta(0), \\ \{q^u, \ldots, Q\}, & \text{if } uv \in \delta^-(0). \end{cases}$$

Likewise, if we consider $\mathcal{L}(a) = \{0, 1, \ldots, Q\}$, for all $a \in A$, and build the AL formulation for the CmVRP given by (3.3b) and (3.2b)–(3.2f) including extra constraints to set trivial low and high loads to zero, we would obtain a formulation very similar to the ones in Fukasawa et al. [74], Mulati et al. [133].

Comparing these formulations to ours, one can note that the preprocessing step in Algorithm 3.1 allows for a possible reduction in the number of variables and constraints. Moreover, the relaxations of our formulations have the potential of providing higher lower bound values, due to removing some arc-load possibilities.

Eventually, we add that it is indeed possible to model the energy limit of the CVRP-LE and CmVRP-LE by adding variables and compact constraints to the AL formulation, similar to which is done, for instance, by Zhang et al. [185]. However, as we are interested in BCP approaches, in Section 3.4 we will focus on models that consider the energy limited by the means of the pricing subproblem.

## 3.3 Review of a Generic BCP Method for VRPs

The BCP approaches that we used in this work is based on the generic BCP method proposed in the work of Pessoa et al. [143]. Therefore, in this section, we focus on outlining the basic components of that method that will be relevant to our work. While the BCP approach in Pessoa et al. [143] was designed to solve VRPs, it has also been applied to other kinds of NP-hard combinatorial optimization problems. An associated

software package, called VRPSolver, has been released, and implements the method's functionalities [156]. We may henceforth refer to this entire framework as VRPSolver.

The approach of VRPSolver is to decompose the problem into an integer master problem (IMP) formulation and a pricing subproblem, where both are coupled to each other in a generic predefined manner. An explicit integer master problem (EIMP) formulation must be provided accordingly to a predefined generic VRP formulation. The definition of the components of the pricing subproblem must be provided in order to be solved as a Resource Constrained Shortest Path Problem (RCSPP). Then the IMP is solved in a BCP fashion, which means that, at each node of the branch-and-bound (BB) tree, a linear programming (LP) restricted master problem (RMP) is iteratively solved via column generation, which relies on the pricing subproblem. Specific cut separation routines may also be called in each resolution of the RMP.

We will start by characterizing the resource constrained paths that are used in the RCSPP for the pricing subproblem. After that, we present the EIMP formulation and the mapping of its variables to arcs of resource constrained paths. Finally, we present advanced features built over these basic components.

### 3.3.1 Resource Constrained Paths

Let $\tilde{D} = (\tilde{V}, \tilde{A})$ be a multi-digraph that will be used to describe the resource constrained paths. The set of vertices $\tilde{V} = \{s, t\} \cup \tilde{N}$ is composed by a source $s$, a sink $t$, and the other vertices in $\tilde{N}$; note that $s$ and $t$ may refer to the same vertex. The set of arcs is given by $\tilde{A}$. Considering $R$ a resource set, we have that for each resource $r \in R$, a *consumption* $\tilde{q}_{ar}$ and an *accumulated consumption interval* $[l_{ar}, u_{ar}]$ are defined for each arc $a \in \tilde{A}$. We may denote $[l_{ar}, u_{ar}]$ as $[l, u]_{ar}$. Consider a non-necessarily elementary path $p = \langle w_0, a_1, w_1, \ldots, w_{l-1}, a_l, w_l \rangle$ of digraph $\tilde{D}$ where $w_0 = s$, $w_l = t$, $\{w_1, \ldots, w_{l-1}\} \subseteq \tilde{N}$, and $l \geq 1$. The resource constrained path $p$ is feasible if it respects the resources consumption, as follows.

Given a resource $r \in R$ and a path $p$ on the digraph $\tilde{D}$, let $\tilde{\rho}_{ir}$ be its amount of accumulated resource consumed at vertex $w_i$, for $0 \leq i \leq l$. It is assumed that $\tilde{\rho}_{0r} = 0$. A resource can be defined as *disposable* or *non-disposable*, depending on how $\tilde{\rho}_{ir}$ is calculated. In the disposable case, the accumulated resource consumption is allowed to be arbitrarily raised, e.g., to fit in the lower bound of an arc's interval. In this case we say that an amount of resource was disposed or dropped. Thus, the accumulated consumption at a vertex $w_i$, where $1 \leq i \leq l$, is given by $\tilde{\rho}_{ir} = \max\{l_{a_i r},\ \tilde{\rho}_{i-1,r} + \tilde{q}_{a_i r}\}$, where $l_{a_i r} \leq \tilde{\rho}_{ir} \leq u_{a_i r}$. The non-disposable case is similar, with the difference that $\tilde{\rho}_{ir} = \tilde{\rho}_{i-1,r} + \tilde{q}_{a_i r}$. We anticipate that the definition of a resource as disposable or non-disposable may have a significant influence on the amount of labels that can be discarded during the algorithm, which directly impacts the performance.

A resource that does not have any negative consumption is said to be *monotone*, otherwise it is *non-monotone*. The set of resources $R$ is divided into the *main* resources $R_M$ and the *secondary* resources $R_s$, where a main resource must be disposable and monotone. The distinct implications of main and secondary resources will be addressed

in Subsection 3.3.4. Let $P$ be the set of all feasible resource constrained paths. Note that the size of $P$ may be exponential on the size of the digraph $\tilde{D}$.

### 3.3.2 Master Problem

The EIMP formulation has the integer variables $[x_j]$ and $[\lambda_p]$. As a modeling product, the $[x_j]$ variables are intended to be directly associated with the arcs of a VRP, where these variables are given by $x_j$ for all $j \in J = \{1, 2, \ldots\}$. The Objective Function (3.4a) must be defined over $[x_j]$ variables. On the other hand, the $[\lambda_p]$ variables are automatically associated with the resource constrained paths $p \in P$. The formulation is presented in the following.

$$\text{(EIMP)} \quad \min \quad \sum_{j \in J} c_j x_j \tag{3.4a}$$

$$\text{s.t.} \quad \sum_{j \in J} \alpha_{ij} x_j \geq \theta_i, \qquad \forall i \in I, \tag{3.4b}$$

$$x_j = \sum_{p \in P} \left( \sum_{a \in M(x_j)} h_a^p \right) \lambda_p, \qquad \forall j \in J, \tag{3.4c}$$

$$L \leq \sum_{p \in P} \lambda_p \leq U, \tag{3.4d}$$

$$x_j \in \mathbb{Z}, \qquad \forall j \in J, \tag{3.4e}$$

$$\lambda_p \geq 0, \qquad \forall p \in P, \tag{3.4f}$$

$$\lambda_p \in \mathbb{Z}, \qquad \forall p \in P. \tag{3.4g}$$

The $[x_j]$ and $[\lambda_p]$ variables are coupled by the constraints (3.4c), which are implicitly defined by the concept of *mapping*, as follows. For each $j \in J$, consider that $M(x_j) \subseteq \tilde{A}$ maps a variable $x_j$ to a non-empty subset of arcs of the digraph $\tilde{D}$. Also consider $M^{-1}(a) = \{j \mid a \in M(x_j)\}$. Let $h_a^p$ denote the number of times an arc $a$ is visited by a resource constrained path $p \in P$. Thus, given a set of resource constrained paths, if an arc $a \in M(x_j)$ is traversed by these paths then this information will be related to the $x_j$ variables, for each $j \in M^{-1}(a)$. Constraints (3.4d) are defined with the usage of $L$ and $U$, intended to be, respectively, the lower and upper bound on the number of paths. We also have the constraints over $[x_j]$ variables in the format of (3.4b), whose indexes are given by $I = \{1, 2, \ldots\}$. As highlighted by Pessoa et al. [143], resource constrained paths can be designed to model intrapath relations while the constraints (3.4b) are suitable to enforce interpath relations.

Once the EIMP formulation is defined, the generic BCP converts that formulation into a version where occurrences of $[x_j]$ variables are substituted by its definition in terms of $[\lambda_p]$ variables, as given by the mapping constraints (3.4c). The formulation is then relaxed generating the LP master problem (MP), stated in (3.5a)–(3.5d). For the sake

of convenience, we present the correspondent dual variables identifiers on the left of the constraints.

$$\text{(MP)} \quad \min \sum_{p \in P} \left( \sum_{j \in J} c_j \sum_{a \in M(x_j)} h_a^p \right) \lambda_p \tag{3.5a}$$

$$(\pi) \quad \text{s.t.} \sum_{p \in P} \left( \sum_{j \in J} \alpha_{ij} \sum_{a \in M(x_j)} h_a^p \right) \lambda_p \geq \theta_i, \quad \forall i \in I, \tag{3.5b}$$

$$(\nu) \quad L \leq \sum_{p \in P} \lambda_p \leq U, \tag{3.5c}$$

$$\lambda_p \geq 0, \quad \forall p \in P. \tag{3.5d}$$

At each node of the BB tree, the LP MP formulation is solved via column generation [17, 59, 72, 123, 48].

Note that the values of the dual variables of the current RMP solution are used to compute the reduced costs. Consider the dual variables of the constraints (3.5b), that are given by $\pi_i$ for all $i \in I$. The reduced cost that corresponds to an arc $a \in \tilde{A}$ is given by:

$$\bar{c}_a = \sum_{j \in M^{-1}(a)} c_j - \sum_{i \in I} \sum_{j \in M^{-1}(a)} \alpha_{ij} \pi_i. \tag{3.6}$$

Let $\nu_-$ and $\nu_+$ be the dual variables of the constraints (3.5c). In the RMP of an iteration of the column generation, the reduced cost of a resource constrained path $p = \langle a_1, \ldots, a_l \rangle \in P$ is denoted by:

$$\bar{c}_p = \sum_{a \in \{a_1, \ldots, a_l\}} \bar{c}_a - \nu_- - \nu_+. \tag{3.7}$$

Observe that if an arc $a$ is traversed more than once in a path $p$ then it will appear repeated with distinct indexes, which plays a role similar to coefficient $h_a^p$ in this operation.

As the interest is on a path with the smallest reduced cost over the current RMP solution, the use of RCSPP from the source $s$ to the sink $t$ is suitable. The RCSPP is solved by a dynamic programming labeling algorithm [143, 158].

Additional cutting planes can be included as extra constraints in the format of (3.5b) after a resolution of the MP. If cuts are separated and added, the MP is re-optimized, otherwise, the MP is considered solved. Note that cuts in this format are *robust* [147], i.e., they do not change the structure of the pricing subproblem, as the dual variables from them are suitable computed in the reduced costs of the arcs via Equation (3.6).

The generic BCP allows for branching over the $[x_j]$ variables and over constraints given by aggregations of $[x_j]$ variables. As both are added as constraints in the format of (3.5b), they are also robust.

### 3.3.3 Packing Sets and Elementarity Sets

The concepts of *packing sets* and *elementarity sets* are pivotal in the context of the generic BCP of Pessoa et al. [143], as the usage of several components rely on definitions of them provided on the problem modeling. One of the benefits is that their definitions are compact and relatively simple. Consider a digraph $\tilde{D} = (\{s, t\} \cup \tilde{N}, \tilde{A})$ and let $\mathcal{P} \subset 2^{\tilde{N}}$ be a collection of packing sets on vertices, where $\mathcal{P}$ is a collection of mutually disjoint subsets of $\tilde{N}$. Note that they cannot contain the source $s$ nor the sink $t$. Each packing set $S \in \mathcal{P}$ indicates that, among the vertices in $S$, *at most one of them can be used in the set of paths of some optimal solution*. In this way, there must exist an optimal solution $(x^*, \lambda^*)$ for the EIMP formulation that satisfies the following constraints:

$$\sum_{p \in P} \left( \sum_{w \in S} h_w^p \right) \lambda_p \leq 1, \quad \forall S \in \mathcal{P}, \tag{3.8}$$

where $h_w^p$ denotes how many times the vertex $w$ is used in a path $p \in P$.

Consider that $\mathcal{E} \subset 2^{\tilde{N}}$ is a collection of elementarity sets on vertices, where $\mathcal{E}$ is a collection of mutually disjoint subsets of $\tilde{N}$. Each elementarity set $S \in \mathcal{E}$ indicates that, among the vertices in $S$, *at most one of them can be used in each path of some optimal solution*. Therefore, there must exist an optimal solution $(x^*, \lambda^*)$ for the EIMP formulation that satisfies the constraints given by:

$$\sum_{w \in S} h_w^p \lambda_p \leq 1, \quad \forall p \in P,\ S \in \mathcal{E}. \tag{3.9}$$

Throughout this paper we use packing and elementarity sets to refer to their versions on vertices. Note that a collection of packing sets is also a collection of elementarity sets, therefore $\mathcal{P} \subseteq \mathcal{E}$. It is not uncommon to have $\mathcal{P} = \mathcal{E}$ composed by singleton sets of the elements in $\tilde{N}$, as done, for instance, for the SymCVRP in Pessoa et al. [143].

Compared to its original packing set form, the usage of packing sets as elementarity sets impose less constraints over the entire solution. At a high level, elementarity sets are useful to impose intrapath relations, while packing sets are useful to impose interpath relations. Elementarity sets are used in *ng*-paths relaxation and path enumeration, which are intended to improve the performance of the RCSPP algorithm used in the pricing subproblem. Packing sets are used, for instance, in the definitions of rounded capacity cuts (RCCs) and limited memory rank-1 cuts (lm-R1Cs), which are applied to the MP.

### 3.3.4 On Pricing Subproblem, Column Generation, and Path Enumeration

In this subsection we focus on advanced features that rely on the values of the dual variables and the employment of elementarity sets.

As previously mentioned, the exact pricing subproblem as a RCSPP is solved by a dynamic programming labeling algorithm [143, 158]. In this approach, a label $L = (u^L, c^L, \tilde{\boldsymbol{\rho}}^{\boldsymbol{L}}, \ldots)$ represents a partial resource constrained path $p^L$ beginning at the source

$s$, ending at the vertex $u^L$, with its current total reduced cost given by $c^L$, and with its current accumulated resources consumption denoted by the components of the vector $\tilde{\boldsymbol{\rho}}^{\boldsymbol{L}}$, which are given by $\tilde{\rho}_r^L$, for all $r \in R$. Additionally, the usage of the last three dots in a label indicates that other fields may be present. Beginning with a label representing a null path from the source $s$, in this search, labels are extended to other labels that represent possible extensions of that partial path. Some labels may be discarded, as follows. Given two labels $L'$ and $L''$ ending at the same vertex, we say that a label $L'$ *dominates* a label $L''$ if any feasible path completion from $p^{L''}$ is also feasible for completing the path $p^{L'}$ and the complete path from $p^{L'}$ has at most the reduced cost of the one from $p^{L''}$. By the resource definitions (see Subsection 3.3.1), note that a precondition for this dominance to occur is that $\tilde{\rho}_r^{L'} \leq \tilde{\rho}_r^{L''}$, for all $r \in R^+$, and $\tilde{\rho}_r^{L'} = \tilde{\rho}_r^{L''}$, for all $r \in R^-$, where $R^+$ and $R^-$ contain the disposable and non-disposable resources of $R$, respectively [143]. Labels identified as dominated are discarded.

We add that this labeling procedure is done bidirectionally, i.e., the search starts also with a label representing a backward partial path that begins in the sink $t$. In this case there is also the concatenation step between forward and backward paths. While this bidirectional search is often faster, it imposes a restriction to the digraph when using non-disposable resources as follows. For each non-disposable resource, its lower and upper bounds on all arcs that enter the sink $t$ must have the exactly same value. Formally, given a value $q'$, the bounds must be defined by $[q', q']_{ar}$ for all $a \in \delta^-(t)$, $r \in R$ [155].

In the generic BCP, labels are grouped into *buckets*. Labels that end at the same vertex and whose accumulated consumption of main resources are within a same predefined interval are stored in the same bucket. Dominance checkings are set to be frequently done only over labels in the same bucket, which is done to speed up computations. Moreover, to avoid a sparse filling of the buckets, the generic BCP supports at most two main resources. A *bucket digraph* is also used to aid efficient dominance checkings between labels in distinct buckets. Such digraph has a vertex per bucket and each arc denotes the possibility of labels extensions between buckets. Bucket arc elimination by reduced cost is also performed [158].

If the RCSPP computes only elementary resource constrained paths, we would have a potentially intractable problem; as it is a strongly NP-hard problem [64]. One option would be the usage of $\mathcal{E}$-elementary paths, where a path is $\mathcal{E}$-elementary if it contains at most one vertex from each elementarity set. Instead, these paths are relaxed into the so-called *ng*-paths [14, 143], described as follows. Let each vertex $w \in \tilde{N}$ have a neighborhood set, dubbed *ng*-set, denoted by $NG(w) \subseteq \mathcal{E}$. Consider an elementarity set $S$ and a *ng*-path that has already visited a vertex in $S$. After this, the path might (re)visit a vertex in $S$ only after visiting a vertex $w$ that does not have $S$ among its neighbors, i.e., $S \notin NG(w)$. Beyond the *ng*-sets initially defined, the generic BCP can dynamically extend them to consider some cycles encountered during the search [32, 153, 143]. We use 8 as the size of the *ng*-sets.

At certain points of the method, the path enumeration procedure tries to enumerate into a pool all $\mathcal{E}$-elementary paths that could participate in an improving solution [13, 46, 137, 143]. Paths can be removed from the pool by standard fixing by reduced costs. An important feature for performance is the discarding of dominated $\mathcal{E}$-elementary partial

paths during the enumeration. As mentioned in Pessoa et al. [143], one way to assure the correctness of this procedure is by satisfying the *sufficient condition for enumeration*. An equivalent form of it is given in the following.

**Sufficient Condition for Enumeration.** Every two feasible partial $\mathcal{E}$-elementary paths $p$ and $p'$ that start in source $s$, end in the same vertex, and visit the same elementarity sets, must have identical coefficients in each essential constraint in (3.5b).

Now we are able to give a glimpse of the column generation algorithm of the generic BCP [143]. Heuristic labeling approaches are tried first, which consist of simplified versions of the exact one. After that, the exact labeling algorithm is executed. The column generation convergence is favored by the usage of the automatic dual price smoothing [140]. If the number of columns exceeds a threshold, a given amount of them with the largest reduced costs are removed. In the sequence, the bucket arc elimination is called after the first column generation converges or when the primal-dual gap decrease since the last call of this elimination procedure surpasses a certain threshold. After this, the path enumeration is called. This enumeration is aborted if the number of partial $\mathcal{E}$-elementary paths or the number of generated paths exceed their respective thresholds. Otherwise, the enumerated paths are stored in a pool, the non-$\mathcal{E}$-elementary paths are removed from the RMP formulation, and future runs of the pricing subproblem are solved by inspecting this pool. Moreover, if the number of enumerated paths becomes smaller than a given threshold, the current RMP formulation, appended with all columns in the pool and integrality constraints, is directly solved by a MILP solver. Depending on time thresholds on each run of the pricing subproblem procedure, after a column generation convergence either cuts are separated or the algorithm perform branching if needed.

### 3.3.5 On Cut Separation

In this subsection we mention advanced components that have direct influence on the LP MP formulation and make use of packing sets. Relying on packing sets, the generic BCP provides the functionalities to separate the classic RCCs [114] and the lm-R1Cs [137].

The generic BCP uses the concept of *RCC separator* to define RCCs [143]. The RCC separator is defined by the specification of a function that associates demands to packing sets, given by $\varphi : \mathcal{P} \cup \{\emptyset\} \to \mathbb{R}_+$, that must provide $\varphi(\{\emptyset\}) = 0$, and a total capacity $\bar{Q} > 0$. Then, the RCCs are generally given by

$$\sum_{p \in P} h^p_{\mathcal{S}} \lambda_p \geq \left\lceil \frac{\sum_{S \in \mathcal{S}} \varphi(S)}{\bar{Q}} \right\rceil, \quad \forall \mathcal{S} \subseteq \mathcal{P},$$

where $h^p_{\mathcal{S}}$ denotes how many times arcs of a path $p \in P$ exits from the collection of packing sets $\mathcal{S} \subseteq \mathcal{P}$. The generic BCP relies on heuristic separation procedures from the CVRPSEP library [125]. A limited number of RCCs, e.g., 100, is separated in each round. Each RCC is robust, therefore, they do not need a special treatment on the pricing subproblem.

Now consider the rank-1 cuts (R1Cs) [143]. Given a collection of packing sets $\mathcal{S} \subseteq \mathcal{P}$, a R1C is given by the application of the Chvàtal-Gomory rounding over the sum of the

inequalities in Equation (3.8) corresponding to the packing sets in $\mathcal{S}$ multiplied by $\rho \in \mathbb{R}_+^{|\mathcal{S}|}$ [138]. This results in:

$$\sum_{p \in P} \left\lfloor \sum_{S \in \mathcal{S}} \rho_S \sum_{w \in S} h_w^p \right\rfloor \lambda_p \leq \left\lfloor \sum_{S \in \mathcal{S}} \rho_S \right\rfloor .$$

Cuts of this type are potentially strong [143], however, as they are non-robust, they have to be especially treated in the pricing subproblem, making it harder. The generic BCP employs the limited memory technique in order to mitigate this issue, leading to the lm-R1Cs [137]. A limited number of lm-R1Cs is separated in each round, e.g., 250, and various sizes of $\mathcal{S}$ (up to 7) may be used in the same round. Exact and heuristic separations are performed.

### 3.3.6 On Primal Heuristic, Strong Branching, and Tree Search Strategy

When finalizing the BB tree root node, our usage of the generic BCP may apply the restricted master heuristic (RMH), that consists of solving the current RMP, added with integrality constraints, directly by a standard MILP solver [157, 143].

The generic BCP uses strong branching, which means that a number of branching candidates are evaluated in order to select the more promising one to be actually branched over [137, 143]. More time is dedicated to strong branching at nodes of lower depth on the BB tree, since these branching decisions tend to have more impact on the overall runtime. Nodes of higher depth rely on the history of previous evaluations to perform the branching The candidates' evaluation involves, for instance, several limited resolutions of the current LP RMP formulation to simulate branching possibilities.

Finally, we point out that the depth-first search (DFS) is the main strategy of exploring the BB tree [156].

## 3.4 Modeling CVRP Variants with Energy Consumption

In this section we describe the modeling, into the generic BCP framework, of basic CVRP variants that take energy consumption into account, namely: CmVRP, CmVRP-LE, and CVRP-LE. First we present some components individually. Next we make use of these building blocks to design consolidated methods for the problems.

### 3.4.1 Resource Constrained Paths Based on Arc-Loads

In this subsection, we propose load and energy resource configurations and the associated descriptor digraphs to be used in the problems of interest. Recall that these components are used to compute resource constrained paths in the RCSPPs of the pricing subproblems. In particular, we define two versions of the load resource: a disposable and a non-disposable. While the standard conduct is the usage of the disposable one [143], there are

problem variants where we can likely profit from the usage of the non-disposable load, as in, for instance, the CVRP-LE. This observation will be more appropriated discussed in Subsection 3.4.5. Besides, to limit the energy consumption of a path, we define an energy resource.

From a problem definition, consider a digraph $D = (V, A)$ and a vehicle capacity $Q$. Also consider a resource constrained paths descriptor digraph given by $\tilde{D} = (\tilde{V}, \tilde{A})$. The set of vertices of $\tilde{D}$ is given by $\tilde{V} = \{0, 0^*\} \cup \tilde{N}$, where: vertex 0 corresponds to the problem's depot and is assigned, at the same time, to the source $s$ and sink $t$; vertex $0^*$ is a placeholder to be defined conveniently; and the vertices in $\tilde{N} = N$ correspond to the problem's customers. Thereby we are dealing with closed resource constrained paths. Let the set of arc-loads be given by $A_{\text{AL}} = \left\{a^\ell \in A \times L \mid \ell \in \mathcal{L}(a)\right\}$, in which the exhaustive set of loads is defined by $L = \{0, 1, \ldots, Q\}$. When we simply substitute the vertex $0^*$ with the depot vertex 0 – which yields $\tilde{V} = \{0\} \cup \tilde{N}$ – and make $\tilde{A} = A_{\text{AL}}$, we obtain which is called the *regular case*, which is used in the subsequent discussion. Moreover, we anticipate that this scheme with a placeholder vertex will be suitable to the presentation of the pre-sink case at the end of this subsection.

Several resources can be associated to a resource constrained paths descriptor digraph $\tilde{D}$. We define a monotone and disposable load resource $\tilde{Q}^+$, which models the load being carried – or equivalently the load capacity being used – across the arc-loads and paths. As a non-disposable counterpart of the previous resource, we also define the monotone and non-disposable load resource $\tilde{Q}^-$. The consumption of each of the resources $\tilde{Q}^+$ and $\tilde{Q}^-$, jointly given by $\tilde{Q}$, at each arc-load is defined by:

$$\tilde{q}_{(uv)^\ell \tilde{Q}} = \begin{cases} 0 & \text{if } uv \in \delta^+(0), \\ q^u & \text{otherwise,} \end{cases} \qquad \forall (uv)^\ell \in \tilde{A}. \tag{3.10}$$

Recover that, while traversing a path, a vehicle is accumulating resource consumption. The accumulated consumption intervals of each of the resources $\tilde{Q}^+$ and $\tilde{Q}^-$, jointly given by $\tilde{Q}$, at each arc-load is denoted by:

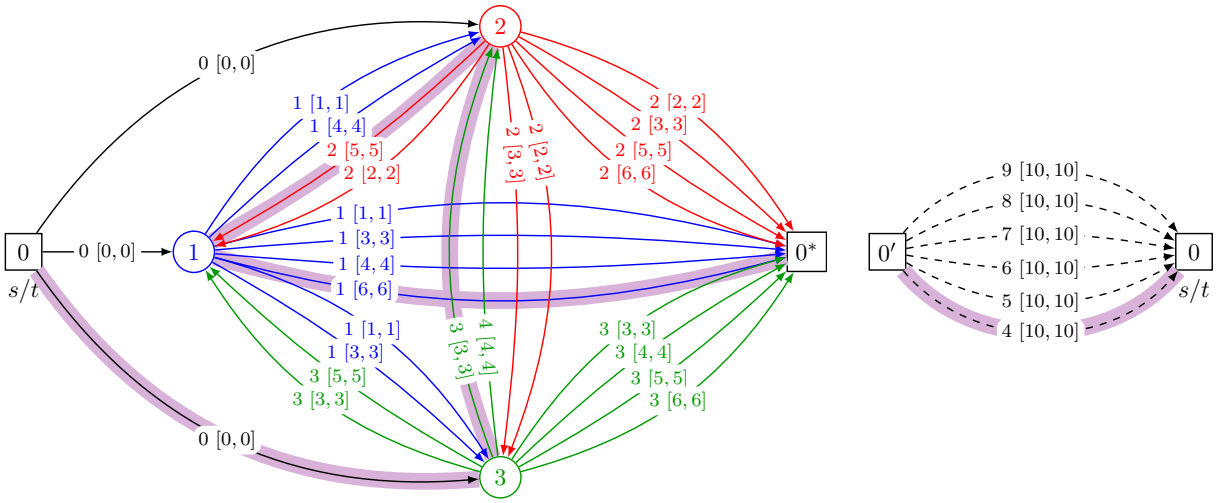
$$[l, u]_{a^\ell \tilde{Q}} = [\ell, \ell], \qquad \forall a^\ell \in \tilde{A}.$$

In order to illustrate this configuration, Figure 3.1 contains a resource constrained paths descriptor digraph $\tilde{D}$ of the small instance S-n04-k1[1], where arc-loads with a load resource is depicted. The regular case is denoted by Figure 3.1a.

For situations that require to limit the energy consumed over a path, as is the case of CmVRP-LE and CVRP-LE, we employ an energy resource, and, for this, consider the energy limit $E$ from the definition of the problems. Thus, we define the energy resource $\tilde{E}$, which is monotone and disposable. This resource models the energy behavior at the arc-loads and paths. The consumption of the resource $\tilde{E}$ at each arc-load is given by:

$$\tilde{q}_{a^\ell \tilde{E}} = d_a \left(q^0 + \ell\right), \qquad \forall a^\ell \in \tilde{A}. \tag{3.11}$$

[1]Instance available online at https://www.loco.ic.unicamp.br/instances




(a) Digraph $\tilde{D}$ with the placeholder vertex $0^*$. The regular case is given whether $0^*$ is replaced with 0, leading the depot 0 to be represented twice in this figure. The pre-sink case happens when $0^*$ is substituted by $0'$ and special arc-loads are added accordingly.

(b) The pre-sink vertex $0'$, the depot 0, and special arc-loads.

Figure 3.1. Resource constrained paths descriptor digraph $\tilde{D}$ of instance S-n04-k1 denoting a load resource. In this case, the demand of each customer matches its number. Each arc-load is labeled in the format $\tilde{q}\ [l, u]$, where $\tilde{q}$ is its load consumption and the interval indicates the bounds on the accumulated load. A feasible path is shadowed.

Observe that the computation of the energy consumption depends on the specific arc-load that it is over, thus the correct visitation of arc-loads must be enforced by other means, e.g., the previously presented load resources. This denotes the pivotal role played by the arc-loads in providing the load information to be used in the computation of the energy consumption. The accumulated consumption intervals of the resource $\tilde{E}$ at each arc-load is denoted by:

$$[l, u]_{a^\ell \tilde{E}} = [0, E], \qquad \forall a^\ell \in \tilde{A}.$$

There are several applicable configurations of the resources. The resource $\tilde{Q}^+$ can be used as main or secondary resource, denoted by $\tilde{Q}^+_M$ and $\tilde{Q}^+_s$, respectively. As it is non-disposable, the resource $\tilde{Q}^-$ can be used only as secondary one, denoted as $\tilde{Q}^-_s$. When the paths are only resource-constrained by load, we can either enable just the resource $\tilde{Q}^+_M$ or we can activate both, what is denoted by $\tilde{Q}^+_M + \tilde{Q}^-_s$. The resource $\tilde{E}$ can be used as main or secondary, written respectively as $\tilde{E}_M$ and $\tilde{E}_s$. When the paths are resource-constrained by both load and energy, we have several possibilities to configure the resources, as showed in Table 3.1. As suggested by Sadykov et al. [158], we defined this multitude of versions to be experimented in order to find suitable resource configurations.

It is worth emphasizing the combined effect of the resources $\tilde{Q}^+_M$ and $\tilde{Q}^-_s$. When the RCSPP procedure – with our definition of load resources – is choosing the next arc-load of a feasible partial path, the non-disposable load resource guarantees that at most one extension is valid over the arc-loads to each possible vertex. Note that this is natural even in the presence of the disposable load resource, as it is less restricted. In the label dominance aspect, consider two labels in the RCSPP procedure. The non-disposable

Table 3.1. Load and energy resource configurations

| Only disposable | With non-disposable |
|---|---|
| $\tilde{Q}_M^+ + \tilde{E}_s$ | $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ |
| $\tilde{Q}_M^+ + \tilde{E}_M$ | $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$ |
| $\tilde{Q}_s^+ + \tilde{E}_M$ | $\tilde{Q}_s^- + \tilde{E}_M$ |

load resource forces that a precondition for one label to dominate the other is that the accumulated load consumption of both have the same value (as stated in Subsection 3.3.4). Observe that this occurs even in the presence of the disposable load resource (again). Conversely, the disposable main resource in scenarios like this potentially improves the performance, as: a main resource might positively influence the dominance checks in the RCSPP procedure (see Subsection 3.3.4); and a non-disposable resource is not allowed to be a main one (see Subsection 3.3.1). The usage of this resources combined are further discussed in subsections 3.4.4 and 3.4.5.

At this point, we deal with the restriction in the usage of a non-disposable resource when the bidirectional search is enabled in RCSPP algorithm, as described in Subsection 3.3.4. For each non-disposable resource, and for $\tilde{Q}^-$ in particular, this caveat imposes that every arc-load entering the sink $t$ must have its lower and upper bounds with the exact same value. To deal with this restriction, we include a special vertex $0'$, called the pre-sink. The customers will have arcs connecting them to the pre-sink vertex instead of the actual sink, while the arcs from the pre-sink to the sink will be tailored to satisfy the mentioned restriction. We say that this digraph is in the *pre-sink case*. We employ the pre-sink case if and only if the non-disposable resource $\tilde{Q}_s^-$ is being used. In the following, we precisely describe the pre-sink version of the digraph.

In the pre-sink case, we start with a resource constrained paths descriptor digraph $\tilde{D} = (\tilde{V}, \tilde{A})$, where the set of vertices is given by $\tilde{V} = \{0, 0^*\} \cup \tilde{N}$ and its meaning is the same as described for the regular case. Thereby, we substitute the vertex $0^*$ with the new special pre-sink vertex $0'$, obtaining $\tilde{V} = \{0, 0'\} \cup \tilde{N}$. Now consider the set $A'_{\text{AL}}$, that corresponds to the set $A_{\text{AL}}$ where each arc-load $(u, 0)^\ell \in \delta^-(0)$ is substituted with $(u, 0')^\ell$. The set of arcs (that are arc-loads) of the digraph $\tilde{D}$ is then given by $\tilde{A} = A'_{\text{AL}} \cup \{(0', 0)^Q \mid \ell \in \mathcal{L}(\delta^-(0'))\}$, where the resource consumption of the special arc-loads from $0'$ to 0 is given by $Q - \ell$, for all $\ell \in \mathcal{L}(\delta^-(0'))$, i.e., the consumption values are the remaining load capacity accordingly to each possibility of accumulated load resource consumption. Therefore, when using the pre-sink case, the load resource consumption previously given in Equation (3.10) must be changed to:

$$\tilde{q}_{(uv)^\ell \tilde{Q}} = \begin{cases} Q - \ell & \text{if } uv = (0', 0), \\ 0 & \text{if } uv \in \delta^+(0), \\ q^u & \text{otherwise,} \end{cases} \qquad \forall (uv)^\ell \in \tilde{A}.$$

A depiction of this case is presented in Figure 3.1, where the pre-sink case includes the Figure 3.1b.

The energy resource consumption, previously given in Equation (3.11), must be adapted for the pre-sink case in order to consider that no energy is consumed over the special arc-loads from $0'$ to 0, as follows:

$$\tilde{q}_{a^\ell \tilde{E}} = \begin{cases} 0 & \text{if } a = (0', 0), \\ d_a \left(q^0 + \ell\right) & \text{otherwise,} \end{cases} \qquad \forall a^\ell \in \tilde{A}. \tag{3.12}$$

Moreover, the correspondence of the arcs from a digraph $D = (V, A)$ to arcs that consider whether one is using the regular or pre-sink case can be given by:

$$\partial(uv) = \begin{cases} (u, 0') & \text{if } uv \in \delta^-(0) \text{ and using the pre-sink case,} \\ uv & \text{otherwise,} \end{cases} \qquad \forall uv \in A.$$

We anticipate that this notation will be useful to obtain simple expressions in the remaining of this paper.

### 3.4.2 Master Problems Based on Arc-Loads or Arcs

We use two main specifications of the EIMP formulation: one that contains only decision variables for arc-loads and another one that contains only decision variables for arcs. Both are mapped to closed resource constrained paths based on arc-loads, where the paths are characterized by a corresponding digraph $\tilde{D}$. Moreover, the lower and upper bounds on the sum of the $[\lambda_p]$ variables are equal to the number of vehicles of the problem, thus, we have $L = U = K$, where $K$ is the number of vehicles in the problems definitions. Note that $L$ and $U$ appear in the Constraint (3.4d) of the EIMP formulation.

The EIMP formulation based on arc-loads is mapped to closed resource constrained paths also based on arc-loads. The formulation is denoted as AL-R. We use the $[y_a^\ell]$ decision variables representing arc-loads given in Equation (3.1). The Objective Function $f(y)$ is chosen according to Equation (3.3), as this model supports minimizing energy consumption or travel distance. The arc-load variables' domain is taken from constraints (3.2e) and (3.2f) of the AL formulation. Note that the $[y_a^\ell]$ variables are instanciations of the $[x_j]$ variables that appear, for example, in the generic Objective Function (3.4a). The customers' out-degree constraints are taken from (3.2c). The latter constraints ensure that each customer will be selected exactly once (via arc-loads) through all the paths, which can be seen as global interpath rule. A constraint on the number $K$ of paths is also given, which play the role of the depot out-degree Constraint (3.2b). The formulation is given in the following.

$$\text{(AL-R)} \quad \min \quad f(y) \tag{3.13a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(w)} \sum_{\ell \in \mathcal{L}(a)} y_a^\ell = 1, \qquad \forall w \in N, \tag{3.13b}$$

$$y_a^\ell = \sum_{p \in P} h^p_{\partial(a)^\ell} \lambda_p, \qquad \forall a \in A,\ \ell \in \mathcal{L}(a), \tag{3.13c}$$

$$\sum_{p \in P} \lambda_p = K, \tag{3.13d}$$

$$y_a^\ell \geq 0, \qquad \forall a \in A,\ \ell \in \mathcal{L}(a), \tag{3.13e}$$
$$y_a^\ell \in \mathbb{Z}, \qquad \forall a \in A,\ \ell \in \mathcal{L}(a), \tag{3.13f}$$
$$\lambda_p \geq 0, \qquad \forall p \in P, \tag{3.13g}$$
$$\lambda_p \in \mathbb{Z}, \qquad \forall p \in P. \tag{3.13h}$$

Each of the $[y_a^\ell]$ arc-load variables are mapped to the correspondent arc-load in the closed resource constrained paths descriptor digraph $\tilde{D}$, as follows:

$$M(y_a^\ell) = \{\partial(a)^\ell\}, \qquad \forall a \in A,\ \ell \in \mathcal{L}(a).$$

By this mapping, the generic coupling constraints (3.4c) are specialized into the coupling constraints (3.13c) of the current formulation.

The branching is performed over expressions given by $\sum_{\ell \in \mathcal{L}(a)} y_a^\ell$, for all $a \in A$, which are aggregations that represent decisions over arcs $a \in A$ of the problem definition. Then, for an arc $a$ selected for branching, it is imposed on one child node that $\sum_{\ell \in \mathcal{L}(a)} y_a^\ell \leq 0$ and on the order that $\sum_{\ell \in \mathcal{L}(a)} y_a^\ell \geq 1$. Recall that these branching constraints are in the format of the generic constraints (3.4b), and therefore, are robust.

The EIMP formulation based on vehicle-flow over the arcs is mapped to closed resource constrained paths based on arc-loads. The formulation is denoted as VF-R. The decision variables denote whether a vehicle pass through an arc, and they are given by

$$z_a = \begin{cases} 1 & \text{if arc } a \text{ is used,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall a \in A.$$

The Objective Function (3.14a) minimizes the travel distance, while the arc variables' domain is enforced by constraints (3.14e)–(3.14f). The customers' out-degree is ensured by constraints (3.14b), which can be seen as interpath constraints guaranteeing that each customer will be selected exactly once in the whole solution. The number of paths that participate in a integer solution is given by the implicitly defined constraint (3.14d). The formulation is presented in the following.

$$\text{(VF-R)} \quad \min \quad \sum_{a \in A} d_a z_a \tag{3.14a}$$
$$\text{s.t.} \quad \sum_{a \in \delta^+(w)} z_a = 1, \qquad \forall w \in N, \tag{3.14b}$$
$$z_a = \sum_{p \in P} \left( \sum_{\ell \in \mathcal{L}(a)} h^p_{\partial(a)^\ell} \right) \lambda_p, \qquad \forall a \in A, \tag{3.14c}$$
$$\sum_{p \in P} \lambda_p = K, \tag{3.14d}$$
$$z_a \geq 0, \qquad \forall a \in A, \tag{3.14e}$$
$$z_a \in \mathbb{Z}, \qquad \forall a \in A, \tag{3.14f}$$
$$\lambda_p \geq 0, \qquad \forall p \in P, \tag{3.14g}$$
$$\lambda_p \in \mathbb{Z}, \qquad \forall p \in P. \tag{3.14h}$$

For each arc $a \in A$, the $z_a$ arc variable is mapped to all the arc-loads related to this arc in the digraph $\tilde{D}$, which are given by $\partial(a)^\ell$ for all $\ell \in \mathcal{L}(a)$. The mapping is given by

$$M(z_a) = \{\partial(a)^\ell \mid \ell \in \mathcal{L}(a)\}, \quad \forall a \in A,$$

which implicitly defines the coupling constraints (3.14c).

The branching is performed over the $[z_a]$ arc variables. When and arc $a \in A$ is selected for branching, the robust constraint $z_a \leq 0$ is imposed on one child node and $z_a \geq 1$ is imposed on the other.

Moreover, it is worth noting that in both formulations, when using the pre-sink case, no variables are mapped to the arc-loads that goes from $0'$ to 0.

### 3.4.3 Packing Sets, Elementarity Sets, and Advanced Features

In this subsection we present some advanced aspects of the usage of the generic BCP. In all of our approaches, the packing sets and elementarity sets are defined as singleton sets over individual customers, as follows:

$$\mathcal{P} = \mathcal{E} = \bigcup_{w \in \tilde{N}} \big\{\{w\}\big\} = \big\{\{1\}, \{2\}, \ldots, \{n-1\}\big\}.$$

Note that the implications of these definitions, via inequalities (3.8) and (3.9), are in conformance with the problems we are dealing with. Recall that the packing sets and elementarity sets are used in the definitions of several advanced components of the generic BCP method.

In our approaches, the *ng*-set $NG(w) \subseteq \mathcal{E}$ of each customer $w \in \tilde{N}$ is initialized containing the elementarity sets corresponding to the closest few (e.g., 8) customers to $w$. As we use path enumeration, we point out that our models satisfy the sufficient condition for enumeration (described in Subsection 3.3.4), as: *(i)* because of the constitution of $\mathcal{E}$, the visitation of the customers is equivalent to the visitation of their elementarity sets; *(ii)* in each formulation, the visitation of each customer $w \in N$ implies in the same coefficient, in any $\mathcal{E}$-elementary path, to be used in the customer $w$'s constraints (3.13b) and (3.14b) converted to the format of the respective specialized MP formulation; and *(iii)* these converted constraints are the only essential ones of their models that fit in the format of constraints (3.5b) of the generic MP formulation.

Besides, we define a RCC separator by the function $\varphi(\{w\}) = q^w$ for all $w \in \tilde{N}$ and $\varphi(\{\emptyset\}) = 0$, and the total capacity of the separator is defined by $\bar{Q} = Q$. It is worth noting that the domain set of $\varphi$ corresponds to $\mathcal{P} \cup \{\emptyset\}$ and the RCC separator definition leads to the classic RCCs.

Finally, the RMH may be enabled in our approaches, which will be detailed per experiment.

### 3.4.4 Consolidated Methods for the CmVRP

In the method to solve the CmVRP we use the EIMP formulation given by AL-R with the objective function of minimizing energy consumption. Note that the arc-load decision variables are necessary to compute the energy consumed on each arc of the EIMP. Two options of resource configuration are used: the load $\tilde{Q}^+_M$ as the main disposable resource, and $\tilde{Q}^+_M + \tilde{Q}^-_s$, that includes a load non-disposable secondary resource. Besides that, we have the option to use the RMH.

Given a constraint $i$ and an arc $a \in A$ of the AL-R formulation, we say that the *pair of constraint and arc* $(i, a)$ *is arc-load uniform* if either every arc-load variable $y^\ell_a$, for all $\ell \in \mathcal{L}(a)$, participate in the constraint $i$ with the same coefficient or either none of them participate. We also say that a *constraint* $i$ *is arc-load uniform* if, for all arcs $a \in A$, the pair $(i, a)$ is arc-load uniform. Note that the constraints (3.13b) on each arc $a \in A$ are applied uniformly over every of its arc-loads variables $y^\ell_a$, for all $\ell \in \mathcal{L}(a)$, thus, these constraints are arc-load uniform. Moreover, observe that all the dynamic added robust constraints imposed by the branching rule and the RCCs are arc-load uniform as well. In this way, for this formulation, the value of the second term of the right-hand side of Equation (3.6) would be the same for all the arc-load variables related to an arc $a \in A$.

Let $\bar{c}_{a^\ell}$ be the reduced cost to be associated to a mapped arc-load $a^\ell \in \tilde{A}$ of the resource constrained paths descriptor digraph $\tilde{D} = (\tilde{V}, \tilde{A})$; the value 0 will be associated to the unmapped ones. Given an arc $a \in A$ of the CmVRP input digraph $D = (V, A)$, the relative differences among the reduced costs $\bar{c}_{\partial(a)^\ell}$ associated to the mapped arc-loads $\partial(a)^\ell \in \tilde{A}$ will be given by the first term of the Equation (3.6), which, in the case of the current formulation, is denoted by $d_a\,(q^0 + \ell)$. This holds as the robust constraints are arc-load uniform. As all the arc-loads of an arc $a \in A$ have the same distance $d_a$, the relative differences will be actually driven by the loads $\ell \in \mathcal{L}(a)$.

The version with only the load resource $\tilde{Q}^+_M$ works correctly, even for non-proven optimal integer solutions. From a label $L = (u^L, c^L, \tilde{\boldsymbol{\rho}}^{\boldsymbol{L}}, \ldots)$, suppose an extension to a vertex $v$ is being evaluated. In this case, extending label $L$ along the arc-load with the lowest feasible bounds of accumulated load interval – i.e., $[\tilde{\rho}^L_r + q^{u^L}, \tilde{\rho}^L_r + q^{u^L}]$, for $r = \tilde{Q}^+_M$ – is always the best choice, as it will yield the lowest possible reduced cost for this arc; leading labels extended over arc-loads with greater load bounds to be dominated. This reasoning resembles what is done for the Cumulative Capacitated VRP by Damião et al. [49], however, related to an arc $a \in A$, they have one arc for each possible number of customers in a route, while we have one arc-load for each possible customers' packageable load demands for this arc. Note that the number of possibilities for arc-loads tends to be much larger than of the number of customers.

The other option of resource configuration for this model is $\tilde{Q}^+_M + \tilde{Q}^-_s$, that uses load as non-disposable secondary resource, in addition to a mandatory load disposable main resource. We also consider the configuration with both resources aiming at comparing their performances on the method.

The latter approach, with $\tilde{Q}^+_M + \tilde{Q}^-_s$, is more similar to the BCP algorithm devised for the CmVRP in Fukasawa et al. [74], as both pricing/RCSPP algorithms are only allowed to build paths that strictly respect the load balance constraints (i.e., there is no possibility

of load excess, a concept that will be further discussed in Subsection 3.4.5). In the following we better compare both CmVRP approaches, as we will present experimental results for the two in Section 3.5. Both are BCP algorithms that rely on a master formulation based on arc-loads and a pricing subproblem solved via RCSPP also based on arc-loads. However, the pricing/RCSPP algorithm of Fukasawa et al. [74] uses an undirected multigraph (while we use a directed one), wherein they compute closed resource constrained shortest paths that are 2-cycle free (we use $ng$-paths). Both approaches separe and dynamically add the robust RCCs. Furthermore, the BCP of Fukasawa et al. [74] also makes use of the following robust cuts (described in Lysgaard et al. [125], Fukasawa et al. [73]): framed capacity, strengthened comb, multistar, partial multistar, generalized multistar, and hypotour. Whereas our approach with the generic BCP can enable the RMH, theirs cannot. In addition, the generic BCP does not allow the use of the cutset branching rule of the BCP of Fukasawa et al. [74]. The BB tree search strategy of both approaches are equal, i.e., by DFS.

### 3.4.5 Consolidated Methods for the CmVRP-LE and CVRP-LE

To devise a method to solve the CmVRP-LE we use the EIMP given by the AL-R formulation with the objective function that minimizes energy consumption. Unlike the method for the CmVRP, we have to assure that the total energy spend by each vehicle to perform its route respects its limit, which is enforced via energy resources on the paths. Thus, there are plenty of correct resource configuration options, as showed in Table 3.1. This model for the CmVRP-LE can be seen as an extension of the one for the CmVRP, thus, we can conclude, via arc-load uniformness of its robust constraints, that the CmVRP-LE model deals correctly with the load balance of the arc-loads in the resource configurations even without the non-disposable load resource $\tilde{Q}_s^-$.

Similar to which is done for the cumulative version, we can present a resolution method for the CVRP-LE. Following this line, the EIMP is defined as the AL-R formulation with its objective function minimizing distance. We use load and energy resources, enabling us to use the resource configuration options given in Table 3.1.

However, there is a drawback in the usage of only disposable load resources for the CVRP-LE, as follows. First, it is not hard to see that the robust constraints are arc-load uniform when minimizing distance. Secondly, the reduced costs associated to the mapped arc-loads $\partial(a)^\ell \in \tilde{A}$ are all exactly the same for an arc $a \in A$ of the CVRP-LE input digraph $D = (V, A)$, as the first term of Equation (3.6) is denoted by the distance $d_a$ for the arc-loads related to arc $a$. In this sense, suppose an extension from a label $L = (u^L, c^L, \tilde{\boldsymbol{\rho}}^{\boldsymbol{L}}, \ldots)$ to a vertex $v$ is in evaluation. As all the arc-loads from $u^L$ to $v$ have the same associated reduced cost, then the dominance rule does not straightforwardly discard arc-loads with accumulated load consumption interval with bounds greater than $[\tilde{\rho}_r^L + q^{u^L}, \tilde{\rho}_r^L + q^{u^L}]$, for $r \in \{\tilde{Q}_M^+, \tilde{Q}_s^+\}$.

One consequence of this drawback is that, given a valid resource constrained path and a load disposable resource $r \in \{\tilde{Q}_M^+, \tilde{Q}_s^+\}$, an arc-load $a_i^\ell$ of it may have load $\ell > \tilde{\rho}_{i-1,r} + q^{w_{i-1}}$. Despite arc-loads permission to violate load balance by an excess like this, these resource constrained paths are still valid for the problem, as the

distance coefficient in the objective function is not affected and the maximum energy limit is respected. Note that even an integer optimal solution can present load excesses. Fortunately, this drawback is saned when we include the non-disposable load resource $\tilde{Q}_s^-$, as denoted in the appropriate column of Table 3.1.

As the CVRP-LE does not need detailed load information of the arc-load variables on the MP formulation, we also designed a method for this problem with the EIMP defined by VF-R, that already minimizes distance. This model has only arc variables at the EIMP, that are mapped to arc-loads of the resource constrained paths descriptor digraph. There are load and energy resources, as stated by configuration possibilities in Table 3.1. Observe that, theoretically, this formulation is not immune to the excess effect with only disposable resources, therefore it can benefit from usage of the non-disposable load resource.

Finally, we make some remarks on presented models with limited energy. A mechanism to ensure the load balance (with excess or not) on the resource constrained paths described by arc-loads is indeed worth using, as the computation of the amount of consumed energy depends on them. It would not be enough, for instance, to remove the load resources and rely only on the RCCs for this, as they do not enforce loads on arcs, despite enforcing minimum numbers of paths over subsets of customers (based on their demands). On the other hand, for the problems with EIMP given by AL-R, it would be enough to use the compact load balance constraints (3.2d) instead of load resources; however, this would very likely result in poor computational performance on a BCP approach.

## 3.5 Computational Experiments

In this section, we present computational results on the approaches discussed, and compare them with the literature or among each other.

### 3.5.1 Implementation, Computational Environment, and Parameters

Our code was implemented with the Julia programming language using VRPSolver[2] 0.4.1 [143] and the standard solver CPLEX[3] 12.10. The experiments were run in a computer with four Intel Xeon Gold 6142 @ 2.60 GHz CPU chips and 252 GiB of RAM. The machine has 64 physical cores that can run at most one thread each and the operating system is the GNU/Linux Ubuntu 18.04. Each test of the experiments were run with only one thread.

The parameters of the VRPSolver in each of our tests are based on the default parameters described in Pessoa et al. [143] (see also Sadykov et al. [158][4]) with few modifications indicated in the following. We set the time limit to 2 h (7200 s). We use the absolute optimality gap tolerance as $10^{-5}$ and artificial variables penalty update factor

[2] https://vrpsolver.math.u-bordeaux.fr

[3] https://www.ibm.com/products/ilog-cplex-optimization-studio

[4] Also considering the VRPSolver default parameters in https://vrpsolver.math.u-bordeaux.fr/doc/parameters.html

of 10 (default is 2). The hard time threshold for the RCSPP labeling algorithm is given by $\tau^{\text{hard}} = 25$ s (default is 20 s). The cut separation tailing-off threshold is given by $\delta^{\text{gap}} = 1.5\%$ (default is 2%). Determining the lm-R1C memory type, we have $\theta^{\text{mem}} = 0$ (default is 2). The maximum number of labels and paths in the path enumeration algorithm are given by, respectively, $\omega^{\text{labels}} = 5 \cdot 10^5$ and $\omega^{\text{routes}} = 5 \cdot 10^6$ (the default values of both are $10^6$). When the RMH is enabled (the default behavior is disabled), we set its maximum time to $\chi^{\text{rm}} = 10$ s. Related to strong branching, we use the tree size estimation ratio of phase 1 given by $\zeta_1^{\text{estim}} = 0.2$ (default is 0.3), candidates number of phase 2 given by $\zeta_2^{\text{num}} = 3$ (already the default value), and tree size estimation ratio of phase 2 given by $\zeta_2^{\text{estim}} = 0.02$ (default is 0.1). The presented parameter values of $\tau^{\text{hard}}$, $\delta^{\text{gap}}$, $\theta^{\text{mem}}$, $\omega^{\text{routes}}$, $\zeta_1^{\text{estim}}$, and $\zeta_2^{\text{estim}}$ match their equivalents described for the SymCVRP in Pessoa et al. [143] and the ones in the SymCVRP example distributed with the VRPSolver. Additionally, the parameter values of $\omega^{\text{labels}}$ and $\zeta_2^{\text{num}}$ coincide with the referred example. Besides, we also changed parameters related to the level of detail of log messages.

### 3.5.2 Instances

The complete set of instances used in our experiments were proposed in previous works; see for instance Uchoa et al. [172]. This set is given by the CVRP/SymCVRP instances of the classes A, B, E, and P, which were obtained from the CVRPLIB[5], and the ones of the class V, which are from the Vehicle Routing Data Sets – COIN-OR[6]. The graphs in these instances are complete digraphs/graphs. Some characteristics of these instances are given in Table 3.2. Columns "Inst" and "#" shows, respectively, the class/instance identifier and the number of instances. Columns "$n$", "$K$", and "$Q$" contains the elements of the sets of, respectively, the number of vertices, the number of vehicles, and the vehicles' capacities. We use a total of 100 instances.

Table 3.2. Characteristics of the instances

| Inst | Type | # | $n$ | $K$ | $Q$ |
|---|---|---|---|---|---|
| A | rand | 27 | 32, 33, 34, 36, 37, 38, 39, 44, 45, 46, 48, 53, 54, 55, 60, 61, 62, 63, 64, 65, 69, 80 | 5, 6, 7, 8, 9, 10 | 100 |
| B | rand-regions | 23 | 31, 34, 35, 38, 39, 41, 43, 44, 45, 50, 51, 52, 56, 57, 63, 64, 66, 67, 68, 78 | 5, 6, 7, 8, 9, 10 | 100 |
| E | cities, rand, rand-mod-e, unkn | 13 | 13, 22, 23, 30, 31, 33, 51, 76, 101 | 3, 4, 5, 7, 8, 10, 14 | 100, 112, 140, 160, 180, 200, 220, 4500, 6000, 8000 |
| P | mod-abe | 24 | 16, 19, 20, 21, 22, 23, 40, 45, 50, 51, 55, 60, 65, 70, 76, 101 | 2, 4, 5, 7, 8, 10, 15 | 35, 40, 70, 80, 100, 115, 120, 130, 135, 140, 150, 160, 170, 280, 350, 400, 3000 |
| V | cities, ody-uly | 13 | 16, 17, 21, 22, 24, 26, 29, 42, 48 | 3, 4, 5 | 5, 6, 7, 8, 9, 10, 11, 15, 16 |
| | | 100 | | | |

[5] http://vrp.atd-lab.inf.puc-rio.br

[6] https://www.coin-or.org/SYMPHONY/branchandcut/VRP/data/index.htm

The column "Type" of Table 3.2 gives a description of how the instances were constructed, in particular: vertices positioned randomly on the plane ("rand"); vertices positioned randomly around points in several regions ("rand-regions"); a map of real world cities ("cities"); randomized modification of instances of class E ("rand-mod-e"); unknown ("unkn"); modifications of instances of the classes A, B, or E ("mod-abe"); and taken from the odyssey of Ulysses ("ody-uly"). All the instances of class V have unitary demands and were converted from Traveling Salesman Problem (TSP) instances of the TSPLIB[7]. It is worth noting that, in this section, column headings in multiple tables may be repeated and, every time a column heading is repeated, it will have the exact same meaning. Therefore, we only explain the meaning of it once.

We use the instances for experiments with the CmVRP, CVRP-LE, and CmVRP-LE and also for a basic experiment with the SymCVRP. For problems considering energy consumption, we define the vehicle curb weight as $q^0 = \lfloor 0.15Q \rfloor$. Comparing with other works on CmVRP, this definition matches what is used in Fukasawa et al. [74] and slightly differs from the usage without rounding of Mulati et al. [133].

We propose a default energy limit definition to be used with the CVRP-LE and CmVRP-LE problems, as follows. Consider an optimal solution for the SymCVRP of an instance. Using the appropriate $q^0$, we calculated the average energy of the routes of this solution, given by $E'$, being careful to always pick the direction of the route that spend less energy. The energy limit for this instance is then given by $E = \lceil 1.1E' \rceil$.

To decrease load-related values, in our implementation, each instance was first preprocessed by having its capacity and customers' demands divided by the greatest common divisor of these values, which is the scaling factor $\gamma$. To keep the models coherent, we substitute $(q^0 + \ell)$ with $(q^0 + \gamma\ell)$ in equations (3.3b), (3.11), and (3.12) and other suitable mentions of this. The only instances with $\gamma > 1$, are E-n13-k4, E-n22-k4, E-n30-k3, E-n33-k4, and P-n22-k8, whose values of $\gamma$ are, respectively, 100, 100, 25, 10, and 100. After preprocessing, E-n23-k3, with $Q' = 4500$, and E-n33-k4, with $Q' = 800$, are the only two instances with $Q' > 400$. It is worth noting that the values of $q_0$ and $E$ are never affected by this scaling procedure. As the data is integral for all instances, the optimal value must be integer.

In order to reduce the number of tests to be done when testing options of methods, we selected a *base sample set* of 19 instances from the complete set, which were selected as follows. For each instance class, we sort the instances in non-decreasing order by number of vertices and then by number of vehicles. The list of each class is then subdivided in groups of five, according to that order, and we select the third instance out of every group; if the last group does not have five instances, we take none. We made small changes to this rule on classes E and V, to make sure that the selection of instances would contain the different types of instances described in Table 3.2. In the end of Subsection 3.5.4 we present experimental evidence that the base sample set can provide results that are similar to the ones of the complete set.

[7] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95

### 3.5.3 Reducing the Number of Arc-Loads and Providing Cutoffs

To aid the performance of the methods, we reduce the number of arc-loads in suitable master formulations and/or resource constrained paths descriptor digraphs by using the procedure showed in Subsection 3.2.2, that computes each arc's discrete load domain as packageable demands of the customers. This provides a reduction on the number of arc-loads of 4.16% on average, compared to only using the elimination of trivial infeasible low and high load values. This latter approached was used in Fukasawa et al. [74] and in our rerun of their code (see Subsection 3.5.4). Despite this percentage being low on average, there are seven instances in which this value is over 10%, namely: E-n13-k4 (48.46% smaller), E-n23-k3 (73.86%), E-n33-k4 (27.37%), P-n16-k8 (51.18%), P-n23-k8 (28.89%), P-n55-k15 (18.68%), and P-n60-k15 (12.05%). The largest relative reduction, in E-n23-k3, went from 2,277,506 to 455,635 arc-loads. After this procedure, the instances with more than 500,000 arc-loads are (in descending order on the number of arc-loads):

1. P-n101-k4 (with 3,719,956 arc-loads),
2. P-n76-k4 (1,764,844),
3. E-n101-k8 (1,719,956),
4. P-n76-k5 (1,371,094),
5. E-n76-k7 (1,033,594),
6. E-n101-k14 (839,956),
7. E-n76-k8 (808,594), and
8. E-n76-k10 (583,594).

Our experiments employ cutoffs to make use of reduced cost fixing possible from the beginning of each method's run and to avoid searching for solutions with too large objective values, favouring smaller BB trees. We recall that the cutoff can improve the usage of other features as, for instance, path enumeration (see the end of Section 3.3 for more details on this). Since the same cutoffs are employed for all methods tested for each problem, we believe that the comparison of the methods is fair. The reasoning for using such cutoffs as an initial input parameter is that we are mainly interested in studying the dual aspect of the methods. If one is also interested in better exploiting/understanding the initial primal bounds computation and the overall computing times considering them, a suitable alternative would be obtaining the upper bounds via fast heuristic approaches.

For each test, the cutoff used is the value of the best known solution (or an upper bound for it) plus one. This is done for the SymCVRP in Fukasawa et al. [73], Uchoa et al. [172]. As feasible solutions for our models and instances always have integer objective values (see Subsection 3.5.2), the extra unit in the cutoff guarantees that the methods have to actually compute at least one feasible solution to prove optimality. As stated by Uchoa et al. [172], this leads to fairer computational comparisons.

In order to obtain upper bounds for the CmVRP instances, we run an experiment with the CPLEX MILP solver using the CmVRP's AL formulation, which is given by the Equations (3.3b) and (3.2b)–(3.2f). This preliminary experiment reported 46 optimal solutions, 51 feasible ones (non-proven optimal), and 3 failures. It yielded the average optimality gap of 4.68% and average time of 4268.9 s, while the average number of processed nodes is above 5400. For the CmVRP-LE, we use a trivial upper bound of $EK$. Since each solution to a CmVRP-LE instance is also feasible for CVRP-LE, we pick the best feasible solution we obtained in a experiment with the CmVRP-LE to compute its distance, denoted by $z^{\mathrm{D}}_{\text{CmVRP-LE}}$. We then use each $z^{\mathrm{D}}_{\text{CmVRP-LE}}$ as upper bound for its respective CVRP-LE instance.

### 3.5.4 Results for the CmVRP

In this subsection, we first present experimental results on BCP algorithms for the CmVRP using the AL-R formulation, where we compare the results of the $\tilde{Q}^+_M$ and $\tilde{Q}^+_M + \tilde{Q}^-_s$ resource configurations over the base set of instances. After that, we compare our best approach with a similar one from the literature, using the complete set of instances. Eventually, we analyze results over the base sample and the complete sets of instances, aiming at showing evidence that the options compare similarly in both instance sets.

#### Evaluating the Resource Configurations

We rely on performance profiles to compare distinct approaches for the same problem. Figure 3.2 presents a performance profile, which shows a plot of the time ratio cumulative frequency, given by $\tau$, by the solved instance ratio, i.e., the proportion of instances solved to optimality. The time ratio of an instance yielded by an approach is given by its computing time divided by the time taken by the fastest approach for the same instance. For more details on performance profiles, see Dolan and Moré [61].

The performance profile in Figure 3.2 compares, over the base sample set of instances, the $\tilde{Q}^+_M$ and $\tilde{Q}^+_M + \tilde{Q}^-_s$ resource configurations of the BCP algorithm for the CmVRP using the AL-R formulation with the RMH enabled. The line of $\tilde{Q}^+_M$ starts with a solved instance ratio of 0.73, while the one of $\tilde{Q}^+_M + \tilde{Q}^-_s$ starts with 0.26. When convenient in this paper, this kind of statement refer to values rounded to a number of decimal places. The former configuration solved to optimality all the 19 instances on this set with a final time ratio of 1.71, i.e., in each of its instance tests, this method is at worst 1.71 times slower than the other. The latter configuration, with a final time ratio of 2.15, solved a total of 18 instances without reporting any feasible solution for the remaining one, B-n51-k7. Note that the line of $\tilde{Q}^+_M$ is always above the one of $\tilde{Q}^+_M + \tilde{Q}^-_s$. Consequently, the performance profile points out that the $\tilde{Q}^+_M$ option outperforms the other approach in this experiment.

We use Table 3.3 to try to better understand how the resource configurations differ, by looking into their average behavior and time composition. This table compares the experimental results of the configurations in the subset of the base set of instances where both have found feasible solutions.

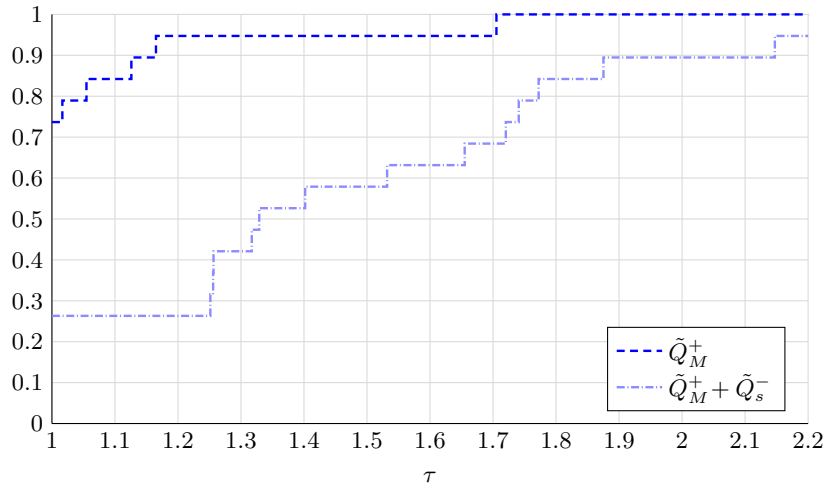



Figure 3.2. Performance profile of the resource configurations $\tilde{Q}_M^+$ and $\tilde{Q}_M^+ + \tilde{Q}_s^-$ of the BCP algorithm for the CmVRP with the AL-R formulation over the base sample set of instances

Besides the first two columns already described for Table 3.2, Table 3.3 has column sequences for each resource configuration. The column "#O" presents the number of proven optimal solutions found and "#S" presents the number of the remaining feasible solutions. The optimality gap is denoted by "G(%)", that is a percentage gap between the value of the best feasible solution found ($z$) and the best lower bound value given by the LP relaxation ($\hat{z}$). The gap's calculation is given by $((z-\hat{z})/z)100$. The root gap denoted by "R(%)" is similar to the previous column, with the only difference being that the lower bound $\hat{z}$ is represented by the value of the LP relaxation at the end of the processing of the root node of the BB tree. The time to solve is given by "T(s)", the time spent in the root node is given by "R T(s)", the time spent in column generation is given in "CG T(s)", and the time taken by reduced cost fixing and path/column enumeration is denoted by "RCF & CE T(s)", all measured in seconds. Note that, throughout this paper, every presented measure of time is rounded to one decimal place. The field "# Nds" denotes the number of nodes processed in the BB tree. The results are row-wise grouped by instance class and the last line presents an overall summary. In these rows, the fields "#", "#O", and "#S" present sums, while the others present averages.

Table 3.3. Comparison of results of the resource configurations $\tilde{Q}_M^+$ and $\tilde{Q}_M^+ + \tilde{Q}_s^-$ of the BCP algorithm for the CmVRP with the AL-R formulation over the base sample set of instances

| | | $\tilde{Q}_M^+$ | | | | | | | | | $\tilde{Q}_M^+ + \tilde{Q}_s^-$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | # | #O | #S | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | #O | #S | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) |
| A | 5 | 5 | 0 | 0 | 0.01 | 90.5 | 71.0 | 1.4 | 63.6 | 8.3 | 5 | 0 | 0 | 0.01 | 121.0 | 103.5 | 1.4 | 99.2 | 5.1 |
| B | 3 | 3 | 0 | 0 | 0 | 159.2 | 159.2 | 1.0 | 50.2 | 85.9 | 3 | 0 | 0 | 0 | 164.9 | 164.9 | 1.0 | 83.8 | 59.1 |
| E | 3 | 3 | 0 | 0 | 0 | 262.0 | 262.0 | 1.0 | 190.7 | 17.7 | 3 | 0 | 0 | 0 | 298.8 | 298.8 | 1.0 | 238.9 | 7.4 |
| P | 4 | 4 | 0 | 0 | 0 | 21.8 | 21.8 | 1.0 | 11.6 | 1.3 | 4 | 0 | 0 | 0 | 39.3 | 39.3 | 1.0 | 30.1 | 0.7 |
| V | 3 | 3 | 0 | 0 | 0 | 1.9 | 1.9 | 1.0 | 0.3 | 0.1 | 3 | 0 | 0 | 0 | 2.0 | 2.0 | 1.0 | 0.7 | 0.0 |
| | 18 | 18 | 0 | 0 | 0 | 100.5 | 95.1 | 1.1 | 60.4 | 19.9 | 18 | 0 | 0 | 0 | 120.0 | 115.1 | 1.1 | 88.1 | 12.7 |

There are some similarities between the resource configurations in the filtered tests of Table 3.3. For instance, both configurations solved all these 18 instances to optimality. Also, they explored only one node each in all the tests except for the one of A-n63-k10, where both explored 3 nodes each and presented the same root gap of 0.04%. Besides, the average time to solve the LPs (not in the table) were rather stable among both, with 0.5 and 0.6 s. On the other hand, some significant differences occur in time-related data. All the average times in T(s), R T(s), and CG T(s) presented for the $\tilde{Q}_M^+$ are lower than their counterparts of $\tilde{Q}_M^+ + \tilde{Q}_s^-$, while all the ones in RCF & CE T(s) are higher in the former. From the configuration $\tilde{Q}_M^+ + \tilde{Q}_s^-$ to the configuration $\tilde{Q}_M^+$, the average column generation time over all instances decreased by a significant amount of 27.7 s, while the RCF & CE T(s) increased by 7.2 s: this difference of 20.5 s comprises almost the entire decrease of 19.5 s in the overall time.

Given the nature of the configurations $\tilde{Q}_M^+$ and $\tilde{Q}_M^+ + \tilde{Q}_s^-$ (see Subsection 3.4.4), it is actually not a surprise that the absence of $\tilde{Q}_s^-$ performs better in these experiments on CmVRP. Supporting this, we observe that the usage of the $\tilde{Q}_M^+$ approach tends to keep extended labels in a similar way of $\tilde{Q}_M^+ + \tilde{Q}_s^-$, however, the former has a more aggressive label dominance rule, discarding labels that the latter does not. Moreover, the presence of $\tilde{Q}_s^-$ led to a smaller average number of generated columns (3348.8 vs. 4423.1), but it was accompanied by a larger number of solved LPs (318.6 vs. 108.3), suggesting a column generation convergence in more rounds.

Finally, we also point out that, in both resource configurations, the class B has, by far, the higher ratio of RCF & CE T(s) in relation to its T(s). One possible explanation is given by the way these instances were constructed, leading to clusters of customers (see Subsection 3.5.2). In instances like this, the algorithm may employ a heavy usage of the reduced cost fixing procedure to remove connections between distinct regions.

### Comparing with the Literature

Now we compare two BCP approaches for the CmVRP using formulations in the line of AL-R: the one proposed in this paper and a rerun of the one of Fukasawa et al. [74], which is, to the best of our knowledge, the state-of-the-art exact method for the problem on the literature. As we are comparing approaches from different origins and implementations, and aiming at having per instance data, we choose to use the complete set of instances in the experiments of both. When convenient, we use FHY16 to refer to the work of Fukasawa et al. [74]. As noted in Subsection 3.4.4, the pricing of FHY16 is roughly similar to our usage of pricing resources given by $\tilde{Q}_M^+ + \tilde{Q}_s^-$, however, as we are aiming at comparing the best approaches, we employed to this end our $\tilde{Q}_M^+$ resource configuration, which is our best one. Moreover, in this comparison we are mainly interested in evaluating the effect of the dual bound quality of both methods, thus our approach is used without (w/o) the primal RMH provided by the VRPSolver, being similar, in this matter, to the BCP algorithm of FHY16. Recall that both use the same cutoffs.

Containing sequences of columns for each approach, Table 3.4 has columns already described for the Table 3.3 and also the column "SB T(s)", which refers to the time spent in strong branching, in seconds. Table 3.4 shows data comparing the experimental results

of the approaches in instance tests where both have found feasible solutions. As stated before, this filtering allows one to focus only on the tests that are directly comparable between the approaches. From the 78 instances considered in the table, all of them were solved to optimality by our $\tilde{Q}^+_M$ w/o RMH approach. The approach of FHY16, on the other hand, reported optimality for 63 of them and feasible solutions for the remaining 15 ones. These measures are naturally reflected on the gap values.

Table 3.4. Comparison of results of BCP algorithms for the CmVRP with formulations like AL-R over the complete set of instances: a FHY16 rerun and the one with $\tilde{Q}^+_M$ resource w/o RMH

| | | FHY16 rerun | | | | | | | | $\tilde{Q}^+_M$ w/o RMH | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | # | #O | #S | G(%) | R(%) | T(s) | R T(s) | # Nds | SB T(s) | #O | #S | G(%) | R(%) | T(s) | R T(s) | # Nds | SB T(s) |
| A | 26 | 23 | 3 | 0.22 | 0.68 | 1,173.5 | 9.0 | 159.7 | 1131.4 | 26 | 0 | 0 | 0.03 | 158.8 | 106.3 | 1.5 | 23.5 |
| B | 20 | 12 | 8 | 1.17 | 1.54 | 3,340.6 | 11.1 | 628.2 | 3184.8 | 20 | 0 | 0 | 0.03 | 326.9 | 217.4 | 1.4 | 19.4 |
| E | 11 | 7 | 4 | 1.27 | 1.51 | 3,309.9 | 27.1 | 152.7 | 3246.4 | 11 | 0 | 0 | 0.03 | 613.8 | 412.9 | 1.5 | 84.9 |
| P | 20 | 20 | 0 | 0 | 0.74 | 364.4 | 7.8 | 102.4 | 328.9 | 20 | 0 | 0 | 0 | 36.2 | 36.2 | 1.0 | 0.0 |
| V | 1 | 1 | 0 | 0 | 1.22 | 35.1 | 9.9 | 13.0 | 31.6 | 1 | 0 | 0 | 0 | 9.6 | 9.6 | 1.0 | 0.0 |
| | 78 | 63 | 15 | 0.55 | 1.04 | 1,808.4 | 11.8 | 262.3 | 1736.3 | 78 | 0 | 0 | 0.02 | 232.7 | 158.8 | 1.4 | 24.8 |

Our approach presents significantly better times in T(s) and SB T(s) in all rows of Table 3.4. More specifically, in relation to the results of the FHY16 rerun, our approach: has an average time of at least 7 times lower; spent at least 13 times more time in the root node; and the average strong branching time takes about an outstanding mark of 70 times lower. These ratios suggest conformance with the VRPSolver efforts of spending more time in the root node and in strong branching [143].

Other measures of interest are the significantly better root gaps that can be seen in Table 3.4. Also, it is worth observing that, while the absolute numbers show that the new approach spends less time doing strong branching, it presents an average strong branching time of 62 s per non-root node, while this average is only 6.6 s in FHY16 (note the massive number of nodes processed by the latter). Therefore, when suitable, the strong branching would favor the selection of more promising nodes to be processed first. Note that, in our approach, the beneficial effect of only using arc-loads that correspond to packageable demands would be spread across several measures, and perhaps, focused on instances where it has more impact. By the results of these experiments, it is clear that the BCP approach with $\tilde{Q}^+_M$ w/o RMH outperforms the one of FHY16.

### State-of-the-Art Results

Considering that the resource configuration $\tilde{Q}^+_M$ was claimed to be the better one over the base sample set of instances and its version without the RMH presented better results than the FHY16 over the complete set of instances, now we present results on the $\tilde{Q}^+_M$ configuration with the RMH enabled over the complete set of instances. We present the results in Table 3.5 and carry on a discussion on these data.

Table 3.5 contains some additional columns/fields beyond ones already presented for previous tables. The column "#N" presents the number of instance tests that finished without reporting any feasible solution, while the column "#X" shows the number of the

remaining tests that were aborted or failed. The tests counted in #N are used in the averages, except for the averages of the gaps. A test counted in #X is not used in any average. Given the proven optimal solutions found for a class of instances, the field "$n'_{\mathrm{O}}$" gives the maximum number of vertices among them. Measured in seconds, the column "LP T(s)" refer to the time spent in solving linear programs. In the grouped or summarized rows, the fields #N and #X contain sums, the field $n'_{\mathrm{O}}$ contain maximums, and the others contain averages. Besides, due to their nature, some detailed time slices may be accounted for more than once, e.g., strong branching time contains part of column generation time and part of the time for solving the LPs.

Table 3.5. Results of the BCP algorithm for the CmVRP using the AL-R formulation with the $\tilde{Q}^{+}_{M}$ resource over the complete set of instances

| Inst | #O | #S | #N | #X | $n'_{\mathrm{O}}$ | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 27 | 0 | 0 | 0 | 80 | 0 | 0.03 | 147.8 | 120.7 | 1.3 | 113.0 | 8.8 | 3.0 | 16.0 |
| B | 22 | 1 | 0 | 0 | 68 | 0.05 | 0.16 | 840.5 | 263.2 | 3.1 | 487.0 | 251.3 | 46.9 | 110.4 |
| E | 11 | 0 | 1 | 1 | 101 | 0 | 0.03 | 1,149.6 | 524.7 | 2.3 | 971.9 | 33.5 | 18.8 | 98.7 |
| P | 22 | 0 | 0 | 2 | 76 | 0 | 0.01 | 198.8 | 108.5 | 1.2 | 143.3 | 21.0 | 1.0 | 41.3 |
| V | 13 | 0 | 0 | 0 | 48 | 0 | 0 | 3.3 | 3.3 | 1.0 | 1.3 | 0.2 | 0.1 | 0.0 |
| | 95 | 1 | 1 | 3 | 101 | 0.01 | 0.05 | 428.2 | 186.0 | 1.8 | 299.9 | 71.0 | 14.5 | 52.2 |

In a first moment, we highlight the efficacy of this configuration: it solved to optimality 95 (out of 100) instances. Hereafter, we will focus on the size of the solved models of each instance in regard to the number of vertices and the number of arc-loads. The instance with the large number of vertices that the configuration was able to optimally-solve is the E-n101-k14 (see column $n'_{\mathrm{O}}$ of Table 3.5), that has 839,956 arc-loads (the 6$^{\mathrm{th}}$ highest in arc-load counting, as presented in Subsection 3.5.3). The instance with the larger number of arc-loads that is optimally-solved is the P-n76-k5, with 1,371,094 (4$^{\mathrm{th}}$). Besides these ones, the others with highest arc-load counting ranked as 5$^{\mathrm{th}}$, 7$^{\mathrm{th}}$, and 8$^{\mathrm{th}}$ were also optimally-solved.

The large number of arc-loads may have a great influence on four out of the five instances that were not solved nor optimally-solved. In this line, the tests of the instances P-n101-k4 (3,719,956, 1$^{\mathrm{st}}$) and P-n76-k4 (1,764,844, 2$^{\mathrm{nd}}$) failed due to an external cause, e.g., the system went out of memory (we emphasize that each memory critical test was run alone). A different situation occurred with the E-n101-k8 (1,719,956, 3$^{\mathrm{rd}}$), whose test ran without issues, but no feasible solution was found. We also uncover that the instance counted in #S is the B-n78-k10, with a gap of 1.08%; which is the one with more vertices and arc-loads (455,392) in class B.

Thereby, with very few exceptions, we stress out the success of the arc-load based approach for moderately-large models for the CmVRP, i.e., with up to 1,371,094 arc-loads. As presented before, the $\tilde{Q}^{+}_{M}$ w/o RMH shows a better performance than the FHY16 approach. As the performance of the $\tilde{Q}^{+}_{M}$ approach (with RMH) is better than the former ones, our approach improves upon the state-of-the-art results for the CmVRP.

Moreover, time measures of features with low values, and thus with low influence on the overall time, are described in the following. The times for cut separation, RMH, and

resolution of column enumerated nodes by the CPLEX MILP solver took general averages of 4.6, 1.4, and 1.6 s, respectively. These times were not reported in Table 3.5. We add that some detailed data values in this table are referred further in this paper, e.g., to make comparisons with other problems. Additionally, detailed experimental results over the base sample set of instances are presented in Appendix B.

### On the Selection of Instances

Beyond the comparisons already made between the resource configurations $\tilde{Q}_M^+$ and $\tilde{Q}_M^+ + \tilde{Q}_s^-$ over the base sample set of instances, we also compare them over the complete set of instances, where the goal is to provide reasonable evidences that our base sample selection is good enough for comparisons. For this, we present the performance profile in Figure 3.3.

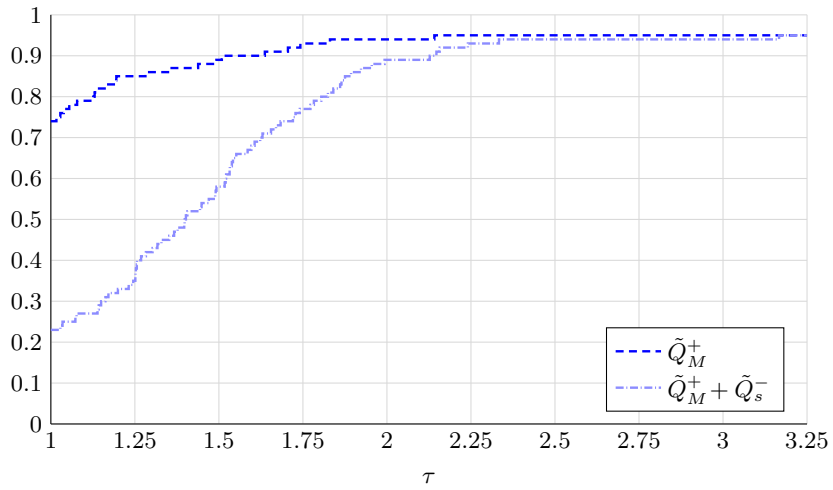



Figure 3.3. Performance profile of the resource configurations $\tilde{Q}_M^+$ and $\tilde{Q}_M^+ + \tilde{Q}_s^-$ of the BCP algorithm for the CmVRP with the AL-R formulation over the complete set of instances

The start of the lines in this performance profile is very similar to the ones for the base sample set of instances, in Figure 3.2. In Figure 3.3, the line of $\tilde{Q}_M^+$ begins with solved instance ratio of 0.74, and the one of $\tilde{Q}_M^+ + \tilde{Q}_s^-$ begins with 0.23. In the end, both ones reached a maximum solved instance ratio of 0.95, with the first with a time ratio of 2.14, while the other reached 3.17. The final of these lines increased (in relation to Figure 3.2) in both configurations, where the $\tilde{Q}_M^+$ for the complete set represents 125% of the one with base sample, while the same measure for $\tilde{Q}_M^+ + \tilde{Q}_s^-$ is 148%. Despite the variations in results between the base and the complete set of instances, both performance profiles (figures 3.2 and 3.3) seem to be in line with each other.

Looking into the overall statistics with the focus on the resource configuration $\tilde{Q}_M^+ + \tilde{Q}_s^-$ with the complete set of instances, we observe that it presented an average time of 569.5 s against 428.2 s of $\tilde{Q}_M^+$ (the latter is also presented in Table 3.5). Unlike the $\tilde{Q}_M^+$ counterpart, no solution was reported on the test of the instance B-n51-k7. On the other hand, an optimal solution was found for the E-n30-k3, which was aborted in

the other attempt. The status of other instances were similar among both. These results also corroborate the conclusions taken over the base sample set of instances.

### 3.5.5 Results for the CmVRP-LE

This subsection delves into the results of the BCP algorithm for the CmVRP-LE using the AL-R formulation with the six resource configurations given by load and energy resources presented in Table 3.1. Compared to the CmVRP, the presence of an energy limit per vehicle/route might enforce a different optimal solution to some instances, as well as it may make instances infeasible (the energy limit is defined in Subsection 3.5.2). As it is a definitive conclusion, a report of infeasibility is considered a success outcome in the performance profiles, in the same way that a solved instance is. Besides, we trace some parallels between the results of the problems with and without limited energy.

#### Evaluating the Resource Configurations

A performance profile comparing the six referred resource configurations over the base sample set of instances is presented in Figure 3.4. All configurations were able to report that the instance E-n31-k7 is indeed infeasible, while none of them reported a feasible solution for B-n51-k7. Note that the latter was the harder one (among the base sample) for our both resource configurations of the BCP algorithm for the CmVRP, whose results we presented in Subsection 3.5.4.

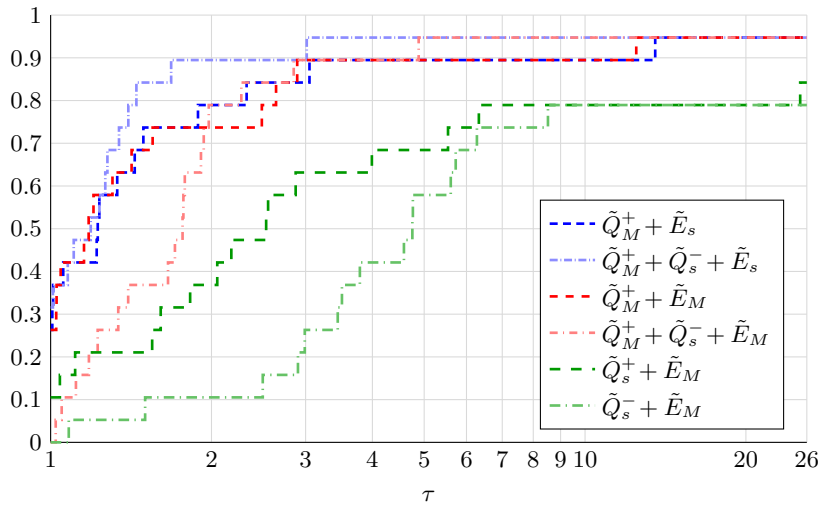



Figure 3.4. Performance profile of the six resource configurations of the BCP algorithm for the CmVRP-LE using the AL-R formulation over the base sample set of instances. The x-axis is in log scale.

We elected two main groups of resource configurations according to the behavior of their lines in the performance profile depicted in Figure 3.4. The best group contains those we evaluated as the four best configurations, that are the ones with a load resource as a main one ($\tilde{Q}_M^+$), while the worst group contains the ones without the load resource as a main one, namely $\tilde{Q}_s^+ + \tilde{E}_M$ and $\tilde{Q}_s^- + \tilde{E}_M$. The best group items are identifiable as the ones who reached a mark of 0.95 (18 instances) in the solved instance ratio. Yet on

the solved instance ratio, we have that the three best starting points of the lines are 0.32, 0.26, and 0.26 for, respectively, $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, $\tilde{Q}_M^+ + \tilde{E}_s$, and $\tilde{Q}_M^+ + \tilde{E}_M$. The time ratio of these points moderately vary, being eventually led by $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$.

The configurations $\tilde{Q}_M^+ + \tilde{E}_s$ and $\tilde{Q}_M^+ + \tilde{E}_M$ presented very similar lines, suggesting that the main (or secondary) status of the energy resource did not have a big impact when only accompanied by the $\tilde{Q}_M^+$. However, their situation changes in opposite ways with the usage of the $\tilde{Q}_s^-$ resource, as follows. The addition of the resource $\tilde{Q}_s^-$ had an improving effect on the $\tilde{Q}_M^+ + \tilde{E}_s$ configuration, yielding the best overall approach for the problem. Conversely, adding $\tilde{Q}_s^-$ to the $\tilde{Q}_M^+ + \tilde{E}_M$ led to a deterioration in its performance, with a very poor start (until the time ratio of about 2).

Due to the fixed number of vehicles of our problems, discovering whether an instance is feasible or not is an important question in the problems with limited energy. Consider the case of B-n51-k7, for which the current CmVRP-LE results don't allow us to know if it is feasible. The question remains open even in the light of some data from an optimal solution on this instance as a CmVRP: the energy consumption of its routes lies in the interval [6677; 12,044] with an average of 8626.57, while the energy limit for this instance is set to 9888. As CmVRP, the total energy of an optimal solution is 60,386, while the best dual bound as CmVRP-LE has value 61,069, obtained by $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$. Therefore, if B-n51-k7 is indeed feasible as a CmVRP-LE instance, it will certainly have an optimal solution distinct from the one as CmVRP.

On an attempt to better understand the average behavior and time composition of the most relevant configurations, we present average statistics for the four best configurations (the ones in the best group) in Table 3.6. In this table, the tests are restricted to the ones in which, for those configurations, either a feasible solution was found or infeasibility was proven; as before in this paper, this restriction aims at focusing on tests that are unequivocally comparable between the contestants. As a matter of fact, from the 18 resulting instances, 17 of them presented optimal solutions while one was proven infeasible.

Table 3.6. Comparison of results of the resource configurations $\tilde{Q}_M^+ + \tilde{E}_s$, $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, $\tilde{Q}_M^+ + \tilde{E}_M$, and $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$ of the BCP algorithm for the CmVRP-LE with the AL-R formulation over the base sample set of instances

| Resource configuration | # | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | # CG | RCF & CE T(s) | LP T(s) | # LP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\tilde{Q}_M^+ + \tilde{E}_s$ | 18 | 0 | 0.31 | 762.3 | 451.4 | 3.0 | 323.7 | 9,902.1 | 389.6 | 15.0 | 445.2 |
| $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ | 18 | 0 | 0.12 | 358.3 | 305.7 | 2.0 | 280.0 | 4,287.1 | 42.3 | 7.9 | 552.3 |
| $\tilde{Q}_M^+ + \tilde{E}_M$ | 18 | 0 | 0.31 | 778.9 | 455.2 | 3.1 | 394.7 | 8,693.1 | 335.4 | 13.5 | 428.5 |
| $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$ | 18 | 0 | 0.11 | 637.0 | 530.0 | 2.6 | 484.0 | 4,010.0 | 107.0 | 16.6 | 615.0 |

Once more we note the effect of using the resource $\tilde{Q}_s^-$, but now from another point of view. The usage of $\tilde{Q}_s^-$ led to the smaller values of root gap and number of explored nodes than the ones without it, as visible in Table 3.6. The $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ resource configuration presents the best computing time average, that corresponds to almost half of the other times. It has the smaller average of explored nodes, however, the instance A-n63-k10 took 7 nodes on it versus 3 in the others. The CG T(s), RCF & CE T(s), and LP T(s) of $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ are the smaller ones among the four. With respect to the

total time of this configuration, the CG T(s) represents 78.1% and the RCF & CE T(s) slice is 11.8%, which are, respectively, the largest and the smallest proportions among the four configurations. The second best time average goes to $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$, however, note this configuration did not have a good performance in Figure 3.4, as previously stated. Despite not being the best, the $\tilde{Q}_M^+ + \tilde{E}_M$ has a good line on the performance profile, however, it presented the worst time average in Table 3.6.

Some statistics do not hold significant high values or do not vary much across these four configurations (not reported in Table 3.6). For instance, the average times for cut separation, RMH, and resolution of column enumerated nodes by the CPLEX MILP solver varies under 6.0 s. Moreover, we have that the average strong branching times per non-root node is 31.0 s and its standard deviation is 6.1 s, thus, not varying much across these configurations.

Considering the performance profile in Figure 3.4 and the data in Table 3.6, here we focus on the two best resource configurations, i.e., $\tilde{Q}_M^+ + \tilde{E}_s$ and $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, to be compared with similar experiments and configurations on the problem without energy limit. Like the method for its counterpart on the CmVRP, the addition of $\tilde{Q}_s^-$ for the CmVRP-LE led to a smaller number of average columns generated (4287.1 vs. 9902.1), a smaller time average for solving the LPs, a larger average number of solved LPs (552.3 vs. 445.2), and a smaller average of RCF & CE T(s). By adding this resource, as a matter of fact, the number of columns decreased 57% in the CmVRP-LE and only 24% in the CmVRP, while the number of LPs increased 24% in the former and (an outstanding mark of) 194.2% in the latter. On the other hand, unlike the CmVRP counterpart, the average time and the column generation time were significantly improved for the CmVRP-LE with the usage of $\tilde{Q}_s^-$.

Note that the CmVRP-LE only difference upon the CmVRP is the usage of a resource to guarantee that the energy limit is respected in the computation of the routes, and particularly, the best CmVRP-LE resource configuration uses the disposable energy resource $\tilde{E}_s$ as a secondary one. Even though the CmVRP-LE's AL-R formulation has arc-load uniform constraints and distinct arc-load objective function coefficients for the same arc (like CmVRP), the usage of $\tilde{Q}_s^-$ (unlike CmVRP), via $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, provided the best results. Therefore, considering the trade-off between creation and discarding of labels, the approach that forbids the creation of labels in which the load resource would be disposed had the best performance, even with the dominance being theoretically weaker in this case. Given the structural similarity of the problems and the difficulties introduced by the $\tilde{Q}_s^-$ resource, we expected that the best results of the CmVRP-LE would be given by the $\tilde{Q}_M^+ + \tilde{E}_s$. It was even more surprising to note the time difference between the approaches for the CmVRP-LE: the average time with $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ is about the half of the average time of the other three configurations in Table 3.6.

Besides that, observe that the energy resource upper bound in the arc-loads might turn some (partial) paths that would be feasible in the CmVRP into infeasible ones in the CmVRP-LE; whereas this mechanism would also avoid the creation of labels with high energy consumption. Moreover, in the best configurations of both (over the base sample set), the average time to solve the CmVRP-LE is three times higher than the CmVRP one, and the average number of nodes of the former is twice the latter's.

### Complete Results and the Effect of the Energy Limit

We report, in Table 3.7, experimental results of the resource configuration $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ for the CmVRP-LE over the complete set of instances, that is our best configuration for this problem. The CmVRP-LE can be seen as simplified versions of the EVRPs studied in Goeke and Schneider [86], Zhang et al. [185], that employ heuristic approaches and rely on instances/models considering several practical technical specifications. We do not compare our results with theirs as we concentrate on using the load-dependent energy consumption on instances derived from the classical CVRP. We point out that this energy consumption is already challenging for BCP approaches, as noted in Section 3.1, and, in addition, some practical technical specifications, e.g., from roads and vehicles, can be incorporated into our approaches in a direct manner.

In Table 3.7, the field "#IF" reports the number of instances reported as infeasible, while the field "$n'_{\text{O/IF}}$" gives the maximum number of vertices of instances with a proven optimal solution found or reported as infeasible. In order to check if a test instance had its optimal solution value changed from the CmVRP to the CmVRP-LE, we rely on our best results for the CmVRP given by Table 3.5. In a determined test, let $z_{\text{CmVRP-LE}}$ be the best solution value of an instance as a CmVRP-LE problem, and let $\hat{z}_{\text{CmVRP-LE}}$ be its best lower bound; we also consider that the value $z'_{\text{CmVRP-LE}}$ equals the former if a feasible solution was found and the latter if only valid lower bound was reported, but not a feasible solution.

Column "$\text{D}^{\text{E}}(\%)$" of Table 3.7 reports the percentage difference of a feasible solution or a lower bound value of instances as CmVRP-LE in relation to solution values of instances as CmVRP ($z_{\text{CmVRP}}$). More precisely, given the solution values of an instance as both problems, their percentage difference is given by $((z'_{\text{CmVRP-LE}} - z_{\text{CmVRP}})/z_{\text{CmVRP}})100$. Column "$\#\text{D}^{\text{E}}$" presents the least number of instances where each optimal solution value was inferred to increase from CmVRP to CmVRP-LE or the instance was concluded to be infeasible. Column "$\#\text{ND}^{\text{E}}$" indicates the number of instances reported with no difference in the solution values, while column"$\#\text{U}^{\text{E}}$" counts the ones with unknown status in this regard. We highlight that no pair of feasible but not proven optimal solutions were obtained/compared.

Table 3.7. Results of the BCP algorithm for the CmVRP-LE using the AL-R formulation with the $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ resource over the complete set of instances

| Inst | #O | #IF | #S | #N | #X | $n'_{\text{O/IF}}$ | $\text{D}^{\text{E}}(\%)$ | $\#\text{D}^{\text{E}}$ | $\#\text{ND}^{\text{E}}$ | $\#\text{U}^{\text{E}}$ | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 27 | 0 | 0 | 0 | 0 | 80 | 1.45 | 21 | 6 | 0 | 0 | 0.12 | 517.7 | 310.3 | 3.0 | 426.4 | 27.8 | 21.5 | 104.0 |
| B | 16 | 4 | 0 | 3 | 0 | 78 | 4.34 | 23 | 0 | 0 | 0 | 0.38 | 1,773.4 | 646.4 | 5.3 | 1,115.0 | 527.1 | 72.0 | 244.3 |
| E | 10 | 1 | 1 | 1 | 0 | 101 | 2 | 8 | 3 | 2 | 0.36 | 0.39 | 2,257.4 | 2127.2 | 1.6 | 1,824.0 | 357.2 | 4.5 | 45.4 |
| P | 18 | 3 | 0 | 2 | 1 | 70 | 0.2 | 9 | 12 | 3 | 0 | 0.01 | 768.5 | 763.1 | 1.1 | 722.3 | 9.1 | 0.8 | 4.5 |
| V | 13 | 0 | 0 | 0 | 0 | 48 | 0.04 | 2 | 11 | 0 | 0 | 0 | 10.1 | 10.1 | 1.0 | 6.6 | 1.5 | 0.2 | 0.0 |
| | 84 | 8 | 1 | 6 | 1 | 101 | 1.67 | 63 | 32 | 5 | 0.05 | 0.16 | 1,029.5 | 692.7 | 2.6 | 783.5 | 179.3 | 23.4 | 92.1 |

As previously mentioned, discovering the feasibility status of the instances is a major concern in the problems with limited energy. In this way, from the complete set of 100 instances, this experiment managed to discover the feasibility status for 93 of them, i.e.,

#O + #IF + #S. We add that none of the instances reported as infeasible are among the 8 ones with more than 500,000 arc-loads.

The larger instances solved are discussed in the following. The instance with the larger number of vertices solved to optimality or reported as infeasible is the E-n101-k14, with 839,956 arc-loads (the $6^{th}$ highest in arc-load counting, according to Subsection 3.5.3). This instance is also the one with more vertices solved by the CmVRP, as can be seen in Table 3.5. The E-n76-k7 (1,033,594, $5^{th}$) is the one with the largest number of arc-loads optimally-solved or infeasible-reported. Also having this outcome, there are the instances ranked as $7^{th}$ and $8^{th}$ in the number of arc-loads. On the other hand, the CmVRP approach has also solved the instance with the $4^{th}$ highest arc-load counting.

The approach had difficulties in some instance models, especially the ones with a very large number of arc-loads. The test of the instance with the $1^{st}$ higher number of arc-loads failed to solve even the first linear program within the time limit, and the ones ranked as $2^{nd}$, $3^{rd}$, and $4^{th}$ in the arc-load counting ended with no feasible solution nor a valid lower bound. Besides that, the only instance reported in #S is the E-n30-k3, with a gap of 3.96% and 119,668 arc-loads; which is distinct from the also unique one reported in #S for the problem version without energy limit.

We compare some measures on the problems with and without energy limit. First, with up to 6.6 s, we mention measures (not in Table 3.7) with no significant impact on the overall average time, namely: average times for cut separation, RMH, and resolution of column enumerated nodes by the CPLEX MILP. These values are similar to ones in the results on CmVRP. Next, we focus on other measures. The summarized data on the last line of the columns G(%), R(%), T(s), R T(s), # Nds, CG T(s), RCF & CE T(s), LP T(s), and SB T(s) for the CmVRP-LE in Table 3.7 are all larger than their counterparts for the CmVRP in Table 3.5. In particular, with limited energy, the averages of total time, column generation time, and RCF & CE time are each more than twice the other one's. Additionally, detailed experimental results over the base sample set of instances are presented in Appendix B.

Another point that might have had an influence in the results is the quality of the cutoffs used. Considering the complete set of instances, after the resolutions, we were able to conclude that the cutoff gap average for the CmVRP is 3.11%, while the cutoff gap average for the CmVRP-LE is 16.76% (note that the identified infeasible instances do not count on the second average).

Besides comparing the performances, we highlight how the energy consumption results changes among the CmVRP and CmVRP-LE problems. The column $D^E$(%) of Table 3.7 shows that the energy consumption average increased 1.67% in relation to the CmVRP, observing that this average is over instances whose tests yielded leastwise a lower bound for the CmVRP-LE. In addition, we have a $D^E$ that is at least 5% in eight instances, wherein B-n56-k7 (with 19.99%), E-n13-k4 (11%), and B-n41-k6 (8.97%) are the three larger ones in this regard. The field #$D^E$, in turn, shows at least 63 instances with a difference in energy consumption, noting that #$D^E$ accounts for the infeasible instances, opposite to the field $D^E$(%). Even with a relatively small average percentage difference between the problems with energy limit and without it, these results' comparisons emphasize the hardness of the CmVRP-LE over the CmVRP with the approaches used. The addition

of the resource $\tilde{E}_s$ is what enforce the energy limit to be respected in each route: it is not a surprise that it imposed a degradation on the performance.

### 3.5.6 Results for the CVRP-LE

The present subsection goes through the experimental results of the BCP algorithm for the CVRP-LE with two distinct EIMP formulations: AL-R and VF-R. Unlike the versions of CmVRP, the usage of VF-R is possible since the objective function of CVRP-LE does not take the consumed energy into account. Each formulation is used with the six resource configurations of load and energy resources given by Table 3.1; hence, there are twelve options to be tested. For each instance, the energy limit per vehicle/route may impose a different optimal solution in relation to the instance as a CVRP problem. Like in the CmVRP-LE results, an instance reported as infeasible is considered as solved in the performance profiles. Throughout this subsection, some comparisons with previous presented problems' approaches are done.

#### Evaluating the Resource Configurations

The performance profile in Figure 3.5 shows twelve lines of experiments over the base sample set of instances. Like in the CmVRP-LE experiments, all options were able to report the instance E-n31-k7 as infeasible and no option was capable of reporting a feasible solution for B-n51-k7. For the latter one, the best dual bound of 1069 was obtained by the options given by AL-R with (w/) $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$, AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$, VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$, and VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$, while its optimal solution value as a SymCVRP is 1032. Thus, if this instance is indeed feasible, it will have optimal solutions distinct from the ones as the problem without energy limit.

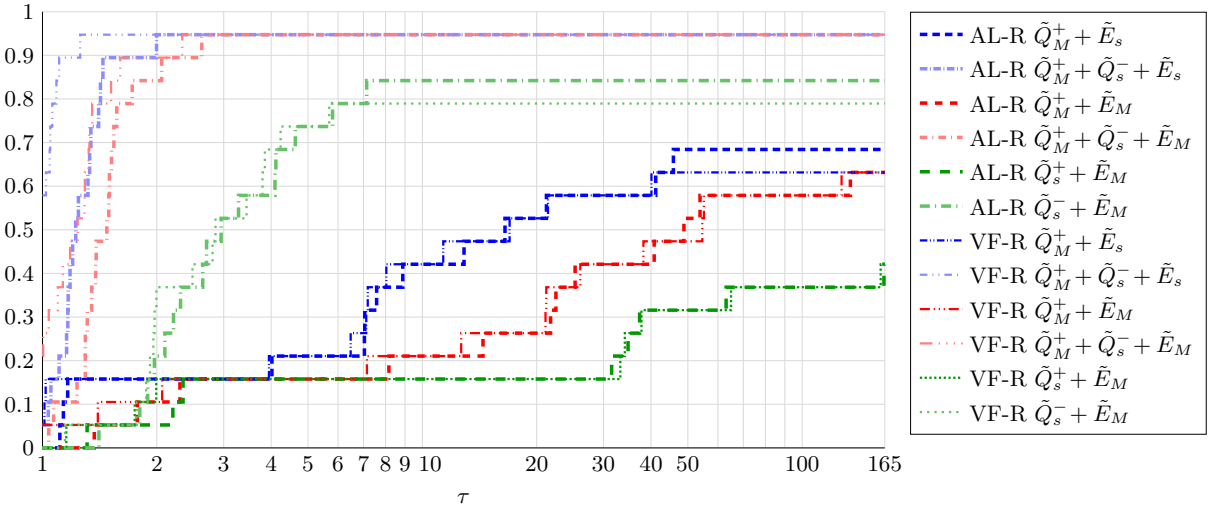



Figure 3.5. Performance profile of the twelve options of the BCP algorithm for the CVRP-LE over the base sample set of instances. The x-axis is in log scale.

Let us identify the options with the best performances. In Figure 3.5 we can see that four options stand out from the rest and reached 0.95 in the solved instance ratio, i.e.,

they optimally-solved or infeasible-reported 18 out of 19 instances. These options are given by AL-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, AL-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$, VF-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, and VF-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$. These four are the only ones with the resources $\tilde{Q}_M^+$ and $\tilde{Q}_s^-$ simultaneously. In the pair with AL-R and in the pair with VF-R, the usage of $\tilde{E}_s$ yielded an undeniably better performance than with $\tilde{E}_M$. Furthermore, in the pair with $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ and in the pair with $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$, the one with the VF-R formulation performed seamlessly better than the one with AL-R.

Note that the VF-R lines have the two best starts in the performance profile, with a solved instance ratio of 0.58 for the $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ resource configuration and 0.21 for $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_M$. As the VF-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ also has the best final point among the lines, with a time ratio of 1.26, it is easy to conclude it is the best approach in this context. Its resource configuration matches the one of the best option for the CmVRP-LE, while the CVRP-LE results favour the VF-R formulation, that is exclusive to it.

Now we focus on the impact of the resources. In the current problem, the AL-R formulation has arc-load uniform constraints and non-distinct arc-load objective function coefficients for the same arc, and besides, due to having only one variable per arc, VF-R mimetizes this aspect. Therefore, the usage of the non-disposable resource $\tilde{Q}_s^-$ has two potential benefits: present an improvement like a similar scenario in CmVRP-LE, and avoid the occurrence of the load excess effect. Recover that the CmVRP-LE's AL-R formulation have distinct arc-load objective function coefficients for the same arc. In the face of this remark, observe that the usage of $\tilde{Q}_M^+ + \tilde{Q}_s^-$ in the CVRP-LE induces a more prominent division in the performance profile plots than in the CmVRP-LE. Besides, the next two better ones in the CVRP-LE are the reminiscent ones including $\tilde{Q}_s^-$.

The six worst options are the ones subject to the load excess effect, due to the absence of $\tilde{Q}_s^-$. Besides, we can see that the performance plots of these options are roughly grouped by pairs with the same resource configuration, namely $\tilde{Q}_M^+ + \tilde{E}_s$, $\tilde{Q}_M^+ + \tilde{E}_M$, and $\tilde{Q}_s^+ + \tilde{E}_M$. In these duos, the VF-R have the best starting points, but not necessarily the best finishing ones, while the intra pair performances along the plots are similar and mixed.

To explore the average behavior and time composition of the best options, Table 3.8 presents average statistics on the four best over the base sample set of instances for all the options considered, in which, for comparison reasons, the tests are confined to the ones who at least a feasible solution was found or infeasibility was proven. Likewise analogous results for the CmVRP-LE, this resulted in 18 instance tests, where 17 of them were optimally-solved and one was infeasible-reported.

Some measures did not occupy a large slice of time, as the average times for RMH and resolution of column enumerated nodes by the CPLEX MILP solver, that are upper bounded by 4.5 s. Other ones did not vary much across the options reported in Table 3.8. Next we give average and standard deviation for several of these measures (not presented in the table). The RCF & CE time took, in average, $69.6 \pm 2.8$ s. The average time to solve linear programs and the average number of them are, respectively, $13.8 \pm 1.1$ s, and $484.9 \pm 8.3$ s. Strong branching average time per non-root node is given by $10.9 \pm 1.7$ s.

Let us try to identify a good option by the data in Table 3.8. From the AL-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ to the VF-R w/ $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$, note the decreasing in the average time

Table 3.8. Comparison of results of the options AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$, AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$, VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$, and VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$ of the BCP algorithm for the CVRP-LE over the base sample set of instances

| Option | # | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | CS T(s) | # CG | # CS |
|---|---|---|---|---|---|---|---|---|---|---|
| AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$ | 18 | 0 | 0.14 | 570.8 | 534.6 | 2.6 | 446.7 | 15.6 | 3,103.1 | 1,154.2 |
| AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$ | 18 | 0 | 0.15 | 710.4 | 668.7 | 2.8 | 573.7 | 15.9 | 4,245.9 | 1,097.9 |
| VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$ | 18 | 0 | 0.14 | 406.8 | 379.0 | 2.6 | 298.6 | 7.6 | 3,717.1 | 1,224.6 |
| VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$ | 18 | 0 | 0.14 | 517.7 | 487.1 | 2.7 | 407.2 | 7.3 | 4,008.6 | 1,128.1 |

of column generation (446.7 to 298.6 s), even with the latter presenting almost 20% more columns. Note as well the decreasing in the average time for cut separation (15.6 to 7.6 s). These differences correspond to almost the total difference in the overall average times of these options. Similar (but larger) times are presented with the usage $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_M$. Despite the average times differences, the average numbers of generated columns and separated cuts did not vary very much.

Therefore, considering Figure 3.5 and Table 3.8, the best option for CVRP-LE combines the smaller number of variables of the VF-R formulation with the benefits of the resource configuration given by $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$, that, for instance, avoids the load excess effect.

Over the base sample set of instances, the average times of the best options to solve the problems CmVRP, CmVRP-LE, and CVRP-LE are, respectively, 100.5 s (AL-R w/ $\tilde{Q}^+_M$), 358.5 s (AL-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$), and 406.8 s (VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$). Using the AL-R for the last one, we obtained 570.8 s, thus, evidencing the value of the VF-R.

### Complete Results and the Effect of the Energy Limit

For the CVRP-LE, the results of the option VF-R w/ $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$ (the best one) over the complete set of instances is given in Table 3.9. In this table, the values of $\mathrm{D}^{\mathrm{D}}(\%)$, $\#\mathrm{D}^{\mathrm{D}}$, $\#\mathrm{ND}^{\mathrm{D}}$, and $\#\mathrm{U}^{\mathrm{D}}$ are defined, changing the superscript, analogous to the ones for the CmVRP-LE in Table 3.7, where one must consider the SymCVRP and CVRP-LE in the place of CmVRP and CmVRP-LE, respectively. Observe that the newly described fields refer to distances instead of energy amounts. In addition, recover that, as noted in Section 3.1, we did not find in the GVRP literature a problem that can accommodate the CVRP-LE as a special case, therefore, we do not compare our results on this problem with the literature.

Table 3.9. Results of BCP algorithm for the CVRP-LE using the VF-R formulation with the $\tilde{Q}^+_M + \tilde{Q}^-_s + \tilde{E}_s$ resource over the complete set of instances

| Inst | #O | #IF | #S | #N | #X | $n'_{\mathrm{O/IF}}$ | $\mathrm{D}^{\mathrm{D}}(\%)$ | $\#\mathrm{D}^{\mathrm{D}}$ | $\#\mathrm{ND}^{\mathrm{D}}$ | $\#\mathrm{U}^{\mathrm{D}}$ | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | 27 | 0 | 0 | 0 | 0 | 80 | 8.42 | 27 | 0 | 0 | 0 | 0.22 | 583.1 | 357.0 | 3.4 | 403.1 | 128.3 | 22.2 | 74.3 |
| B | 12 | 4 | 0 | 7 | 0 | 67 | 13.06 | 23 | 0 | 0 | 0 | 0.53 | 2,619.0 | 815.5 | 9.5 | 1,608.7 | 721.4 | 210.5 | 377.6 |
| E | 10 | 1 | 0 | 2 | 0 | 101 | 6.06 | 12 | 1 | 0 | 0 | 0.43 | 2,346.5 | 1699.6 | 3.2 | 2,034.5 | 202.7 | 20.0 | 197.1 |
| P | 18 | 3 | 0 | 3 | 0 | 70 | 2.26 | 17 | 4 | 3 | 0 | 0.03 | 1,023.4 | 849.1 | 1.3 | 938.5 | 11.7 | 2.7 | 9.4 |
| V | 13 | 0 | 0 | 0 | 0 | 48 | 5.36 | 13 | 0 | 0 | 0 | 0.18 | 194.1 | 59.9 | 2.2 | 108.0 | 66.9 | 11.7 | 9.9 |
|  | 80 | 8 | 0 | 12 | 0 | 101 | 7.4 | 92 | 5 | 3 | 0 | 0.24 | 1,335.7 | 716.5 | 4.1 | 982.6 | 238.4 | 59.2 | 136.1 |

In order to evaluate the performance of the method, we report the number of feasibility statuses discovered of and the size of the larger solved instances. This BCP option for the problem was able to optimally-solve 80 instances and declare 8 ones infeasible, while the remaining 12 ones (completing the 100 test instances) fell into the status of no solution found. As expected, the infeasible instances are the same ones reported by experiments on CmVRP-LE, however, note that the ones with no solution reported there summed only 7 ones. The solved instance with most vertices is the E-n101-k14, that is also the one with more arc-loads among the solved ones; where we recall that this instance has 839,956 arc-loads, which is the $6^{\text{th}}$ highest in this matter. The next solved ones with more arc-loads are those ranked as $7^{\text{th}}$ and $8^{\text{th}}$ in this regard. In comparison, the best CmVRP-LE approach solved the $5^{\text{th}}$ while the CmVRP's solved the $4^{\text{th}}$ with highest arc-load counting.

In Table 3.9, the field $\text{D}^{\text{p}}(\%)$ denotes that the average solution distance increased 7.4% comparing with the SymCVRP; wherein the field $\#\text{D}^{\text{p}}$ denotes that at least 92 instances presented a distance difference. We also note that there are eight instances with $\text{D}^{\text{p}} \geq 15\%$, where the ones with larger values are B-n56-k7 (with 48.51%), E-n13-k4 (26.72%), and B-n78-k10 (26.04%). On the other hand, the CmVRP-LE presented solution energy differences in at least 63 instances, in spite of the instances of both using the same energy limit.

The times with a small slice of the average overall time are given by the average times for cut separation, RMH, and resolution of column/path enumerated nodes by the CPLEX MILP solver, that are, respectively, 7.4, 2.5, and 1.5 s; wherein the first and last ones are similar to results on CmVRP-LE, while the RMH took more time in the CVRP-LE. Now we move onto the measures with large slices of time. From the CmVRP-LE to the CVRP-LE, the summarized time averages increased. The overall average time increased 30% (+306.2 s), the column generation time increased 25% (+199.1 s), the RCF & CE time is 33% (+59.1 s) larger, and the time to solve linear programs is about 153% (+35.8 s) larger. The average strong branching time is 48% larger, however, due to the higher number of nodes, the average strong branching time per non-root note went from 57.6 to 43.9 s. We add that this happened even with the CVRP-LE using cutoffs with an average gap of 4.67% in relation to the results obtained, which is more than three times smaller than the same measure for the CmVRP-LE. Additionally, detailed experimental results over the base sample set of instances are presented in Appendix B.

The average time of the method for the CVRP-LE being higher than the one for the CmVRP-LE raises two points. First, the values of $\text{D}^{\text{p}}(\%)$ and $\#\text{D}^{\text{p}}$ of both might indicate that the chosen energy limits have constrained more the optimal solutions of the CVRP-LE than the ones of the CmVRP-LE, resulting in harder to compute solutions on the CVRP-LE. Secondly, it may be associated to the intrinsic hardness of the CVRP-LE, as its objective function minimizes distance and the new resource constrained the energy, while in the other, the objective function and the new resource both deal with energy.

## 3.6 Concluding Remarks

In this paper we devised BCP approaches for three CVRP variants that minimize the amount of load-dependent energy consumption and/or impose energy limit per route. For each RCSPP (corresponding to a pricing subproblem) related to these problems, we used resource constrained paths descriptor digraphs based on arc-loads. In these digraphs, the usage of the arc-load concept is pivotal in tracking and limiting the energy consumption throughout each route. In an attempt to improve performance, we employed a preprocessing procedure to only keep the arc-loads corresponding to packageable demands, which led to, in average, 4.16% less arc-loads than a previous approach in the literature.

For the CmVRP, we formulated the EIMP by the AL-R formulation, where the best results are yielded with the resource configuration given by $\tilde{Q}^{+}_{M}$, i.e., solely using the load disposable resource. The central components of the method are, in a sense, similar to the ones of the BCP of Fukasawa et al. [74], however, the dominance rule among the labels is more aggressive in the former approach. Comparing the approaches under similar conditions, in the average, we were able to improve the optimality gap from 0.55% to 0% in a time at least 7 times lower (over 78 instances for which both reported at least a feasible solution). Thus, we improved upon the state-of-the-art exact results for this problem.

The CmVRP-LE, proposed in this paper, differs from the CmVRP precisely in the energy limitation per vehicle. The EIMP formulations used for both are very similar, while the resource configuration of choice for the CmVRP-LE is given by $\tilde{Q}^{+}_{M}+\tilde{Q}^{-}_{s}+\tilde{E}_{s}$, where non-disposable load ($\tilde{Q}^{-}_{s}$) and energy ($\tilde{E}_{s}$) resources are present. The presence of $\tilde{Q}^{-}_{s}$ means that, in this case, the best performance was achieved by the approach that forbids the creation of labels in which the load resource would be disposed to the detriment of discarding these labels by the dominance rule.

The last problem solved in this paper, also proposed by us, is the CVRP-LE. Due to the fact that its objective function does not consider energy consumption, we were able to use an EIMP formulation given by VF-R, that has variables per arc instead of, the more numerous, variables per arc-load (as used in AL-R). The best resource configuration is given by $\tilde{Q}^{+}_{M}+\tilde{Q}^{-}_{s}+\tilde{E}_{s}$, where the presence of $\tilde{Q}^{-}_{s}$, in this case, avoids an important performance issue intrinsic to this problem: the load excess effect.

Over a set of 100 instances, the best average times to process the CmVRP, CmVRP-LE, and CVRP-LE are, respectively, 428.2 s, 1029.5 s, and 1335.7 s; which probably indicates a reverse order of hardness among these problems. Besides, we can solve instances with up to 101 vertices in each of these problems. It is also worth noting that, for the majority of the instances tested, the variants with limited energy resulted in optimal solutions different from their counterparts without energy limit. Amid the VRPs that consider load-dependent energy consumption, we recall that the enforcing of a limit on the energy consumption on the routes of the vehicles by the means of the pricing subproblem constitutes a novelty in the BCP/BP literature.

Hereafter we point out some possibilities of future research. Directly related to the results obtained, it is worth investigating a more fundamental reason that led the BCP with the $\tilde{Q}^{+}_{M}+\tilde{Q}^{-}_{s}+\tilde{E}_{s}$ resource configuration to yield the best results on the CmVRP-LE,

as this resource option was not expected to be the best for this problem. Another point regards the number of arc-loads: the instance with the highest number of arc-loads that we are able to solve in this paper contains 1,371,094 of them. In this sense, our results suggest that the approaches should be improved to be applied to instances that generate higher number of arc-loads. One option of research in this line is to improve the removing arc-loads, by relying, for instance, in the energy limit and branching decisions. Another possibility is to investigate the usage of cuts specifically devised for arc-load-base formulations, as the ones of Pessoa et al. [141].

Some extended versions of the problems dealt with in this paper might be suitable to be solved by BCP approaches similar to the ones we used. These extensions may consider: the effect of speed variations on the energy consumption, the usage of time windows, and the usage of recharging or refueling stations. In particular, it is worth noting that, even with the energy consumption being generally nonlinear on the speed [19, 75], arc-load approaches offer the potential to consider speed in a linear fashion for the CmVRP, CmVRP-LE, and CVRP-LE (when the distance is not interpreted as time), as, in these cases, the optimal speed per arc and load can be pre-computed and incorporated in the costs of arc-loads. On the other hand, the inclusion of time windows in this scenario leads to a more complex situation, as the speed values impact on the time taken to traverse each arc, and thus in the ability to reach vertices within theirs windows [19].

We deal with distance-based and energy-based objective functions. One way to directly consider both criteria is to add them together, where we can adjust the trade-off between them by weighting each criterion. However, the precise weight of each criterion might not be available beforehand and it may be hard to measure distinct criteria in the same unit. In this sense, a more flexible approach would be to solve bi-objective variants of the problems minimizing both previous criteria, thus the decision-maker – using the Pareto front [33] – would be able to make an informed decision on the trade-off between the criteria after the optimization. Once practical technical specifications are incorporated into the models (e.g., from vehicles and roads [19]), the first criterion could represent time while the second one could denote energy consumption or greenhouse gas emissions. Previous works in this line for GVRPs encompasses the exact approach of Kabadurmuş et al. [105] and the heuristic approaches of Demir et al. [56], Erdoğdu and Karabulut [69].

In addition, the consideration of BCP approaches for these problems with a heterogeneous fleet might be of interest in environmental-concerned applications that have a mix of vehicles with distinct types of energy sources, e.g., with electric, alternative fuel, fossil fuel, and/or hybrid vehicles engines. Allowing a variable number of vehicles is also a relevant practical aspect, that may as well arise in a VRP with heterogeneous fleet. Besides, it may be worth employing BCP approaches and energy considerations to the VRP with Drones, that is a related problem where each truck carries several drones which, by turn, are dispatched from the truck to perform deliveries [175, 148, 160, 176, 126, 29].

# Acknowledgments

The author M.H. Mulati acknowledges the financial support of Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná (FA), Secretaria de Estado da Ciência, Tecnologia e Ensino Superior do Paraná (SETI), Governo do Estado do Paraná, and Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES). M.H. Mulati had scholarship of / *foi bolsista da* CAPES / Programa de Doutorado Sanduíche no Exterior (PDSE) / Processo nº 88881.190068/2018-01. The author also acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). The author R. Fukasawa acknowledges the support of the NSERC [funding reference number RGPIN-2020-04030]. This research is partially supported by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) procs. 425340/2016-3 and 314366/2018-0, and Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) procs. 2015/11937-9 and 2016/01860-1.

# Chapter 4

# Tighter Analysis of an Approximation for the Cumulative Vehicle Routing Problem

**Abstract.** We deal with the Cumulative Vehicle Routing Problem (CmVRP), a generalization of the Capacitated Vehicle Routing Problem, which objective is to minimize the fuel consumption. Gaur et al. [80] gave a 4-approximation based on a well-known partition heuristic to the Traveling Salesman Problem. We present a tighter analysis obtaining a $\left(4 - \frac{4}{3s^3Q^2}\right)$-approximation, where $Q$ is the capacity of the vehicle and $s$ is a scaling factor. To the best of our knowledge, this is the best proved approximation for the CmVRP so far.

## 4.1 Introduction and Previous Work

The Cumulative Vehicle Routing Problem (CmVRP) was proposed by Kara et al. [108]. The objective is to minimize the fuel consumption, given that the fuel consumed by distance unit is linearly proportional to the total weight being carried (vehicle + load).

An instance of the CmVRP is defined in the following. A complete undirected graph $G = (V, E)$ with vertices $V = \{0, 1, \ldots, n\}$, where 0 is the depot and the other vertices are customers. There is an object of weight $w_i \in \mathbb{Q}_{>0}$ for each customer $i$, and we consider that $w_0 = 0$. Each edge $uv \in E$ has a length $d_{uv} \in \mathbb{Q}_{>0}$ satisfying the triangular inequality. An empty vehicle with capacity $Q \in \mathbb{Q}_{>0}$ and weight $W_0 \in \mathbb{Q}_{>0}$ is initially located at the depot, and also, the weight of an object does not exceed $Q$. In a feasible solution $S$, we have that the only vehicle is repeatedly used in $k$ directed cycles, each one including the depot, to form a schedule that picks up the objects at the customers and drops them in the depot, visiting every customer exactly once. The objective is to obtain such a schedule that minimizes the fuel consumed.

Let $\mu > 0$ be a constant that relates how much fuel is consumed by weight per distance unit. We define $a = \mu W_0$ and $b = \mu$. The fuel consumed by the vehicle to traverse the cycle $C_j$ is $f(C_j) = a|C_j| + b\sum_{i \in C_j} w_i d_i^S$, where $|C_j|$ is the length of the cycle $C_j$, and $d_i^S$ is the distance traveled by the vehicle carrying the object from being picked in the

customer $i$ until being dropped at the depot in the schedule $S$. The fuel consumed by a vehicle to perform a schedule $S$ is $f(S) = a\sum_{j=1}^{k}|C_j| + b\sum_{i=1}^{n} w_i d_i^S$. Let $d_i$ be the shortest distance between vertex $i$ and the depot.

In this paper, we roughly follow the structure of Gaur et al. [80]. For our contribution, we use the theorems 1–4 from Gaur et al. [80] and the definitions presented below.

**Theorem 4.1** (Haimovich and Rinnooy Kan [96], Gaur et al. [80]). *Let $C^*$ denote an optimal Traveling Salesman Problem (TSP) tour of the graph $G = (V, E)$. Then, the total distance traveled by a vehicle to bring all objects to the depot is at least* $\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right)$.

A *subtour* is a TSP tour that visits a subset of $V(G)$, and, when it is clear, we will use tour to denote it. W.l.o.g., consider that the vertices of a tour $C$ are numbered as $0, 1, \ldots, n$ in the order that they appear in the sequence, and 0 is the depot. $|C|$ is the length of the tour $C$. By a *cluster* $[i, j]$ we mean a set of a sequence of vertices in the tour $C$ from $i$ to $j$, with the extremes included. Considering $k \geq 2$, and $1 < x_1 < x_2 < \cdots < x_{k-1} \leq n$, a *cluster partition* denoted by $P = [1, x_1, x_2, \ldots, x_{k-1}, n]$ of tour $C$ is a decomposition of $C$ into the $k$ clusters $[1, x_1]$, $[x_1, x_2], \ldots, [x_{k-1}, n]$. From a cluster partition $P$ of $C$, we are able to construct $k$ subtours $C_1, \ldots, C_k$ such that: to traverse the subtour $C_j$, the vehicle begins at the depot, visits the vertices of $C_j$ in increasing order, and ends in the depot. The length of $P$ is given by $l(P) = |C_1| + \cdots + |C_k|$.

**Theorem 4.2** (Altinkemer and Gavish [4], Gaur et al. [80]). *Let an integer $s > 0$ be a scaling factor in a way that $sw_i$, for every $i \in V$, $sW_0$, and $sQ$ are positive integers, such that $sw_i \leq sQ$ for every $i \in V$. Let $C$ be a TSP tour of $G$ and let $Q$ be the vehicle capacity. Then, there exists a cluster partition $P$ of $C$ with total length at most* $4\frac{\sum_{i=1}^{n} w_i d_i}{Q} + \left(1 - \frac{2}{sQ}\right)|C|$.

**Theorem 4.3** (Gaur et al. [80]). *Let $C^*$ be an optimal TSP tour, and let $Q$ be the capacity of the vehicle. Then, the minimum fuel consumed by the vehicle to bring all objects to the depot is at least* $a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\left(\sum_{i=1}^{n} w_i d_i\right)$.

**Theorem 4.4** (Gaur et al. [80]). *Let $\beta > 0$ be a positive rational number, $C$ be a TSP tour, and assume that the vehicle has infinite capacity. Then, there exists a cluster partition $P = [1, x_1, x_2, \ldots, x_{k-1}, n]$ of $C$ with total fuel consumption at most* $\left(1 + \frac{2}{\beta}\right) b\left(\sum_{i=1}^{n} w_i d_i\right) + \left(1 + \frac{\beta}{2}\right) a|C|$.

## 4.2 Our Contribution

We provide, in theorems 4.5 and 4.7, a refined analysis of the algorithm of Gaur et al. [80], showing a tighter approximation ratio than the one presented by them.

**Theorem 4.5** (From Gaur et al. [80] with a tighter bound). *Let $\beta > 0$ be a positive rational number, $C$ be a TSP tour, and $Q$ be the vehicle capacity. Then, there exists a cluster partition $P = [1, x_1, x_2, \ldots, x_{k-1}, n]$ of $C$ with total fuel consumption at most* $\left(1 + \frac{2}{\beta}\right) b\left(\sum_{i=1}^{n} w_i d_i\right) + \left(1 + \frac{\beta}{2}\right) a|C| + 4a\frac{\sum_{i=1}^{n} w_i d_i}{Q} - 2a\frac{\sum_{j=1}^{k}|C_j|}{sQ}$.

*Proof.* Considering infinite capacity, there exists a cluster partition $P$ of tour $C$ with fuel consumption $f(P)$ with an upper bound given by Theorem 4.4. Let $C_1, C_2, \ldots, C_k$ be the subtours corresponding to the cluster partition $P$. Let $W_j$ be the total weight of the objects picked by the vehicle in the subtour $C_j$. If $W_j \leq Q$, then $C_j$ satisfies the capacity restriction and, consequently, we keep the cluster corresponding to the subtour $C_j$ with fuel consumption $f(C_j)$ unchanged. On the other hand, assume that $W_j > Q$: by Theorem 4.2 there exists a refined cluster partition $P_j$ of $C_j$ such that the total weight of the objects in each cluster of $P_j$ is at most $Q$, and there exists an upper bound on its length.

Now, given a subtour $C_j$, we will give an upper bound on the fuel consumption $f(P_j)$. W.l.o.g., we assume that the fuel consumption $f(C_j)$ of the subtour $C_j$ is obtained by a traversal in clockwise order (the reversed case is symmetric). Consider that the vehicle traverses each subtour $C_{jl}, 1 \leq l \leq k_j$ in the partition $P_j$. We have that $V_j$ is the set of vertices of the tour $C_j$. Consider that, for each vertex $i \in V_j$, we have $d_i^{C_j}$ that represents the distance traveled by the vehicle from picking object $i$ to dropping it at the depot in tour $C_j$, and analogously, $d_i^{P_j}$ represents the respective distance for the cluster partition $P_j$. We have that $d_i^{P_j} \leq d_i^{C_j}$, because the cluster partition $P_j$ is a refinement of the tour $C_j$. Thus we can write $f(P_j) = \sum_{l=1}^{k_j} f(C_{jl}) = \sum_{l=1}^{k_j} \left( a|C_{jl}| + b \sum_{i \in C_{jl}} w_i d_i^{P_j} \right) = a \sum_{l=1}^{k_j} |C_{jl}| + b \sum_{i \in C_j} w_i d_i^{P_j} \leq a l(P_j) + b \sum_{i \in C_j} w_i d_i^{C_j} \leq 4a \frac{\sum_{i \in C_j} w_i d_i}{Q} + a|C_j| - a|C_j| \frac{2}{sQ} + b \sum_{i \in C_j} w_i d_i^{C_j} = f(C_j) + 4a \frac{\sum_{i \in C_j} w_i d_i}{Q} - a|C_j| \frac{2}{sQ}$, where we used the upper bound on $l(P_j)$ given by Theorem 4.2.

We consider that $P'$ is the final cluster partition that includes all the clusters, the ones unchanged as well as the ones refined. Thus, the total fuel consumption is given by $f(P') = \sum_{j=1}^{k} f(P_j) \leq \sum_{j=1}^{k} \left( f(C_j) + 4a \frac{\sum_{i \in C_j} w_i d_i}{Q} - a|C_j| \frac{2}{sQ} \right) = \sum_{j=1}^{k} f(C_j) + 4a \frac{\sum_{j=1}^{k} \sum_{i \in C_j} w_i d_i}{Q} - 2a \frac{\sum_{j=1}^{k} |C_j|}{sQ} = f(P) + 4a \frac{\sum_{j=1}^{n} w_i d_i}{Q} - 2a \frac{\sum_{j=1}^{k} |C_j|}{sQ}$. By the upper bound of Theorem 4.4 on $f(P)$, we have that $P'$ is a cluster partition satisfying the theorem. □

**Lemma 4.6.** *Let $C^*$ be an optimal TSP tour in a complete graph $G = (V, E)$ with weight function $d$ that are part of an instance of the Capacitated Vehicle Routing Problem. Then, $|C^*| \leq 2 \sum_{i=1}^{n} d_i$.*

*Proof.* Recall that $|V| = n + 1$, with the vertices numbered from 0 (depot) to $n$. By definition $|C^*| = \sum_{uv \in E(C^*)} d_{uv} \leq \sum_{uv \in E(C^*)} (d_u + d_v) = \sum_{i \in V(C^*)} 2d_i = 2 \sum_{i=0}^{n} d_i = 2 \sum_{i=1}^{n} d_i$, where we used: that $d_{uv} \leq d_{u0} + d_{0v}$ for $uv \in E(C^*)$ by the triangular inequality, the definition that $d_{i0} = d_{0i} = d_i$ for every $i \in V$, the fact that each vertex $i$ is an extreme of exactly two edges of $C^*$, and the definition $d_0 = 0$. □

**Theorem 4.7.** *There exists a factor $4 - \frac{4}{3s^3Q^2}$ polynomial-time approximation algorithm for the CmVRP.*

*Proof.* Given as input an instance of CmVRP as previously described, consider the algorithm with the steps: *(i)* compute a tour $C$ of $G$ by the well-known Christofides' algorithm [80], which guarantees that $|C| \leq \frac{3}{2}|C^*|$; *(ii)* compute a cluster partition $P^*$ of tour

$C$ with optimal fuel consumption by a dynamic programming algorithm in time $O(n^2)$ as done in Gaur et al. [80]; and *(iii)* return the subtours $C_1^*, C_2^*, \ldots, C_k^*$ of $P^*$.

In this algorithm, we optimally calculate $P^*$ of $C$ in polynomial time. The analysis is being made over a heuristic algorithm that also calculates a cluster partition $P$ of a tour, thus we can state that $f(P^*) \leq f(P)$. Let $S^*$ be an optimal routing scheduling in fuel consumption. By theorems 4.5 and 4.3, we have the ratio

$$
\begin{aligned}
\frac{f(P^*)}{f(S^*)} &\leq \frac{\left(1+\frac{2}{\beta}\right) b \left(\sum_{i=1}^{n} w_i d_i\right) + \left(1+\frac{\beta}{2}\right) a|C| + 4a\frac{\sum_{i=1}^{n} w_i d_i}{Q} - 2a\frac{\sum_{j=1}^{k}|C_j|}{sQ}}{a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\sum_{i=1}^{n} w_i d_i} \\
&\leq \frac{4\left(a\left(\frac{|C^*|}{2} + \frac{2}{2}\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\sum_{i=1}^{n} w_i d_i\right) - 2a\frac{\sum_{j=1}^{k}|C_j|}{sQ}}{a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\sum_{i=1}^{n} w_i d_i} \qquad (4.1) \\
&\leq 4 - \frac{2a\sum_{j=1}^{k}|C_j|}{sQ\left(a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\sum_{i=1}^{n} w_i d_i\right)} \\
&\leq 4 - \frac{2a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right)}{sQ\left(a\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + b\sum_{i=1}^{n} w_i d_i\right)} \qquad (4.2) \\
&= 4 - \frac{2\mu W_0\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right)}{sQ\left(\mu W_0\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + \mu\sum_{i=1}^{n} w_i d_i\right)} \qquad (4.3) \\
&\leq 4 - \frac{2W_0\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right)}{sQ\left(sW_0\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + sW_0\sum_{i=1}^{n} sw_i d_i\right)} \qquad (4.4) \\
&= 4 - \frac{2\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right)}{s^2Q\left(\max\left(|C^*|, 2\frac{\sum_{i=1}^{n} w_i d_i}{Q}\right) + \sum_{i=1}^{n} sw_i d_i\right)} \\
&\leq 4 - \frac{4\frac{\sum_{i=1}^{n} w_i d_i}{Q}}{3s^3Q\sum_{i=1}^{n} w_i d_i} \qquad (4.5) \\
&= 4 - \frac{4}{3s^3Q^2}.
\end{aligned}
$$

To obtain (4.1), we chose $\beta = \frac{2}{3}$ and used that $|C| \leq \frac{3}{2}|C^*|$. We used Theorem 4.1 to get (4.2). To obtain (4.3), recall that $a = \mu W_0$ and $b = \mu$. In (4.4), we made use of the fact that $s$ and $sW_0$ are integers. To obtain (4.5), we deal with the absolute value of the fraction: we kept or lower the numerator; and we majored the denominator applying the fact that $|C^*| \leq 2\sum_{i=1}^{n} d_i \leq 2\sum_{i=1}^{n} sw_i d_i$ as Lemma 4.6 states and as $sw_i$ are integers for every $i \in V$, and the fact $2\frac{\sum_{i=1}^{n} w_i d_i}{Q} = 2\frac{\sum_{i=1}^{n} sw_i d_i}{sQ} \leq 2\sum_{i=1}^{n} sw_i d_i$, as $sQ$ is integer. □

# Acknowledgments

Supported by CNPq (311499/2014-7 and 425340/2016-3), FAPESP (2015/11937-9), and SETI-PR / Fundação Araucária / CAPES.

# Chapter 5

# A Robust Optimization Approach for the Vehicle Routing Problem with Selective Backhauls

**Abstract.** The Vehicle Routing Problem with Selective Backhauls (VRPSB) aims to minimize the routing costs minus the revenues collected at backhaul customers optionally selected. In this paper, we explore a VRPSB under uncertain revenues, where a deterministic VRPSB is formulated as a mixed-integer linear programming problem and a robust counterpart is derived. We devised a branch-and-cut algorithm to solve the robust optimization (RO) formulation and compared it with a heuristic approach. Besides, we estimate probabilistic bounds of constraint violation for this formulation. The RO approach studied showed high potential to tackle the problem, requiring similar computing effort and maintaining the same tractability as the deterministic modeling.

## 5.1 Introduction

The Vehicle Routing Problem with Backhauls (VRPB) is an extension of the classic Vehicle Routing Problem (VRP), where two different types of customer are considered in the network: linehaul customers, those who receive goods from a depot (outbound logistics), and backhaul customers, those who send goods back to the depot (inbound logistics). Integrated inbound-outbound logistics planning can reduce the distance of empty trips, which are responsible for significant transportation costs [122], thus increasing the efficiency of both vehicles and routes, and reducing fuel consumption and emissions of air pollutants [149].

One approach whereby the VRPB includes an optional selection of backhauls is the VRP with Selective Backhauls (VRPSB) [12]. In this type of problems, the backhauls are selected according to the revenues they provide. The first application of the VRPSB is described in Yano et al. [181] to solve a real transportation problem of a company operating in a retail chain. The company owned a fleet of vehicles and had to plan delivery routes from the distribution center to some stores and pickup routes from vendors to the

distribution center. Since the private fleet could not cover all the vendors, due to capacity limitations, the company outsourced carriers to pickup the material from the remaining ones. Therefore, one of the decisions of the problem was to select which backhauls to visit. By optimizing the VRPSB, the company was able to save a considerable amount of money.

Several successfully exact approaches applied to solve VRPs rely on cut and column generation algorithms [146, 15, 143]. These exact methods were also employed to solve both VRPB [90, 152] and VRPSB. Gutiérrez-Jarpa et al. [92] formulate a VRPSB as a mixed-integer linear programming (MILP) model and use a branch-and-cut (BC) algorithm to solve it. Granada-Echeverri et al. [90] modeled a VRPB with a MILP formulation based on characteristics of an Open VRP, allowing to obtain several new best known solutions for instances of Goetschalckx and Jacobs-Blecha [87] and Toth and Vigo [169].

Due to efficient computational performance, heuristics are the most used algorithms to solve the majority of VRPB and its variants [167]. For the VRPSB, García-Nájera et al. [77] proposes a Similarity-Based Selection Multi-Objective Evolutionary Algorithm (SSMOEA) to solve a multi-objective problem considering minimization of number of vehicles, travel distance and uncollected revenue. The metaheuristic was tested for adapted instances of Goetschalckx and Jacobs-Blecha [87], providing solutions for instances with up to 150 customers in the range of 1000 to 8000 s.

Notwithstanding the practical application of the VRPSB to real-world problems (e.g., parcel services), few works have been described in the literature that use this variant of the VRP [92]. This statement is also valid for other similar problems such as the VRP with Deliveries and Selective Pickups (VRPDSP). The VRPDSP often arises in reverse logistics, where a vehicle delivers a request to a customer and may collect an amount of material in the same location, resulting in a profit [168, 150, 31, 43]. Thus, in the VRPDSP, customers requiring deliveries are also those requiring pickups.

In the VRPSB, selecting one backhaul rather than another depends highly on the revenue it provides and its impact in mitigating routing costs. For instance, the revenue may be related to the quality of products to pick up or the suppliers may have several products available with different quality attributes [184, 5, 6]. However, to the best of our knowledge, only Allahviranloo et al. [3] reported the development of a selective VRP model with uncertainty in revenues, showing evidence of the advantages of addressing uncertainty in humanitarian logistics. More surprisingly, a recent literature review on the VRPB research shows that no work has yet investigated this problem under uncertainty, which is usually present in practice [109].

To deal with uncertainty in VRPs, two main approaches stand out in the literature: stochastic programming and robust optimization (RO) [11, 22]. Stochastic programming is a well-known technique used when the probability distribution of the uncertain parameters are known, and the goal is to optimize the expected value of a solution while maintaining its feasibility for the scenarios considered [27]. When no knowledge on the probability distribution of the uncertainty is available, robust optimization emerges as a suitable alternative [37, 91]. In this case, the goal is to obtain a solution that is robust (i.e., feasible) against all possible uncertainty realizations. Robust optimization is, in fact, the most recent trend to deal with uncertainty in optimization problems [91].

The present paper analyses the robustness of routing plans obtained under revenue uncertainty for a robust VRPSB, where it is required to achieve a minimum revenue from collected raw-materials. The interest on studying the uncertain revenues derive from two main aspects: *(i)* a deviation in the quality of raw-materials can impact the efficiency of the productive processes (at the depot) and, consequently, the production costs of a company, and *(ii)* for large distances, and particularly for daily routes, the deviation in the expected revenue may turn the predefined routes very costly for consecutive days, which may cause significant financial damage to the company. Particularly for the wood-based supply chain, the quality of raw-materials is becoming more and more important, since, in one hand, it allows to differentiate suppliers [6] and, on the other hand, it impacts the efficiency of subsequent manufacturing processes [184, 5]. In addition, although several characteristics of raw-materials can be easily obtained, such as diameter and length, others can only be roughly estimated, such as moisture.

Over the VRPSB, we explore the aspects of RO and its respective budget of uncertainty [23], devised an exact solution method, in the form of a BC algorithm, and compared it with a heuristic approach based on the Adaptive Large Neighborhood Search (ALNS). The ALNS was adapted to the robust VRPSB by Santos et al. [159], the complete version of this paper. In addition, we study different methods to derive bounds for the probability of constraint violation of the RO model. The main contributions of this paper are as follows.

1. A formulation for the VRPSB under revenue uncertainty, contributing simultaneously for two gaps in literature, given by the lack of works dealing with uncertainty in VRPBs and uncertain revenues in VRPs; and

2. A BC algorithm embedded with RO to solve the problem under uncertainty, that is able to effective solve medium-sized instances.

The remainder of this paper is organized as follows. The deterministic formulation, as well as the RO model for the VRPSB are detailed in Section 5.2, where the methods to derive probabilistic bounds are also defined. In Section 5.3, both solution methods used in this work are presented, which are respectively a BC algorithm and an ALNS metaheuristic. The results are presented and discussed in Section 5.4, particularly regarding the robustness of the model under uncertainty, the structure of the solutions, and computational performance of both solution methods. Finally, conclusions and insights of this work are stated in Section 5.5.

## 5.2 Problem Description and Formulations

In this section, a MILP formulation for the deterministic VRPSB is presented first. The deterministic model is then reformulated with the proper robust counterpart in order to incorporate uncertainty in revenues. Afterwards, several different methods of estimating probabilistic bounds are presented to estimate the probability of constraint violation.

### 5.2.1 Deterministic VRPSB

The problem can be defined by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a fleet of $K$ vehicles. Set $\mathcal{L} = \{1, \ldots, n\}$ comprises the linehaul customers with demands $q_i > 0$ for all $i \in \mathcal{L}$. Set $\mathcal{B} = \{n+1, \ldots, n+m\}$ comprises the backhaul customers with revenues $p_i > 0$ for all $i \in \mathcal{B}$. The depot is partitioned into two vertices: an origin depot 0 and a destination depot $n+m+1$. We anticipate that a split depot allows a better adaptation of the problem to be solved in the BC algorithm, as discussed in Subsection 5.3.1. Finally, the set of all vertices in the network is $\mathcal{V} = \{0, n+m+1\} \cup \mathcal{L} \cup \mathcal{B}$. The edges in $\mathcal{E}$ are given by $\{(0,j)\colon j \in \mathcal{L}\} \cup \{(i,j)\colon i \in \mathcal{L} \cup \{n+m+1\},\ j \in \mathcal{L} \cup \{n+m+1\},\ i < j\} \cup \{(i,j)\colon i \in \mathcal{L},\ j \in \mathcal{B}\} \cup \{(i, n+m+1)\colon i \in \mathcal{B}\}$ with costs $c_{ij} \geq 0$ for all edges $(i,j) \in \mathcal{E}$. Note that, for the sake of a clean notation we are using the notation $(i,j)$, $i,j$, and $ij$ to represent an (undirected) edge given by $\{i,j\}$. Set $\mathcal{K} = \{1, \ldots, K\}$ represents the fleet of identical vehicles, each one with a capacity $C > 0$, all of them initially located at the (origin) depot. A route is a sequence of vertices given by $r = \langle i_0, i_1, i_2, \ldots, i_s, i_{s+1} \rangle$ with $i_0 = 0$ and $i_{s+1} = n+m+1$ and with $\{i_1, i_2, \ldots, i_s\} \subseteq \mathcal{L} \cup \mathcal{B}$. The total cost of a route is given by the sum of its costs minus its revenue.

The objective in the VRPSB is to find a set of at most $|\mathcal{K}|$ routes (tours) minimizing the total cost in such a way that: *(i)* all linehaul customers are visited, each one in exactly one route; *(ii)* all the linehaul customers of a route must be visited before the backhaul customer (if any); *(iii)* there is no route with just one customer if it is a backhaul; *(iv)* the sum of the demands of the linehaul customers in a route is at most $C$; and *(v)* a minimum revenue obtained from the raw-materials picked up at backhaul customers is attained. Four additional features of real world transportation problems are assumed for the problem: *(vi)* the revenue associated to each backhaul customer is a function of external factors not related to route planning; *(vii)* each route visits at most one backhaul customer; *(viii)* backhaul customers have enough raw-material available to supply the depot with multiple vehicles; and *(ix)* backhaul customers can only be visited after all deliveries are performed (precedence constraint). These assumptions derive from the fact that many inbound routes are supported by full truckload vehicles that, in a single visit to a backhaul customer, collect an amount of load that equal their capacity. Example of such cases are found in the manufacturing industry (e.g., forests with large supply for mills) and in retail (e.g., supplier inbound operations to the retailers' distribution centers). In particular, for wood-based supply chains the inbound transport is often carried in full truckloads (e.g., Carlsson and Rönnqvist [34], Hirsch [100], Derigs et al. [57]). It is also worth mentioning that: woods parks do not usually present capacity problems to accommodate several types of raw-materials; and the depot is able to receive the raw-material collected by all vehicles. Finally, the precedence constraint is an important aspect in practice because: *(a)* vehicles are often rear-loaded and this precedence constraint avoids the rearrangement of the loads at linehaul customers [62, 112], and *(b)* linehaul customers have higher service priority than backhaul customers [154, 171] because they have time-windows necessary to inspect the received loads.

To better illustrate the network of the VRPSB, an example of a feasible solution is given in Figure 5.1. In this example, the network is composed of seven linehaul customers

(white circles), five backhaul customers (gray circles) and a depot (partitioned into two squares). Four different routes are created. Two vehicles visit backhaul customers after visiting the linehaul customers and two other vehicles only visit linehaul customers, returning then back empty to the depot.

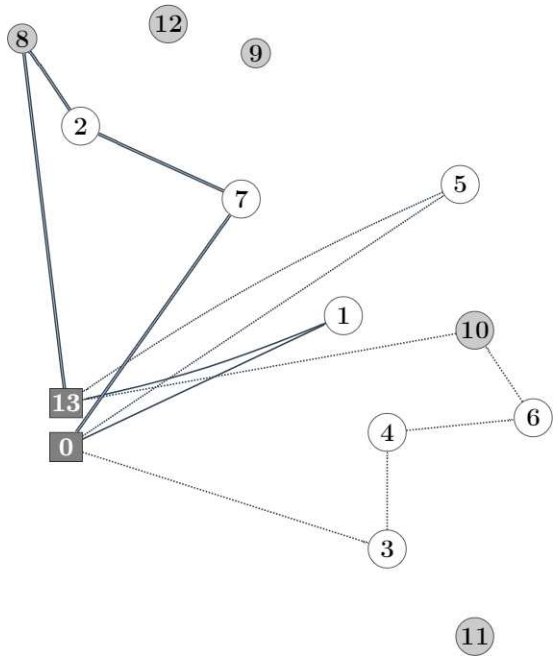



Figure 5.1. Example of a feasible solution for a VRPSB instance with $\mathcal{V} = \{0, 13\} \cup \mathcal{L} \cup \mathcal{B}$, where $\mathcal{L} = \{1, 2, 3, 4, 5, 6, 7\}$ and $\mathcal{B} = \{8, 9, 10, 11, 12\}$. Omitting costs, demands, and revenues. The divided depot is given by the squares and $\mathcal{K} = \{1, 2, 3, 4\}$.

The deterministic VRPSB is presented as a two-index vehicle-flow formulation using the following decision variables: $x_{ij}$, for all $(i, j) \in \mathcal{E}$, that represents the number of times the edge $(i, j)$ is used in a tour, and $\kappa$, that denotes the number of distinct paths that begin at the origin depot 0 and end at the destination depot $n + m + 1$. The following notations are also used to formulate the problem: $\delta(X, Y) = \{(i, j) \in \mathcal{E} : i \in X,\ j \in Y\}$, $\delta(S) = \{(i, j) \in \mathcal{E} : i \in S,\ j \notin S\}$, $\delta(X, i)$ corresponds to $\delta(X, \{i\})$, and $\delta(i)$ is equivalent to $\delta(\{i\})$. The deterministic VRPSB reads as follows:

$$\min \sum_{(i,j)\in\mathcal{E}} c_{ij}x_{ij} - \sum_{j\in\mathcal{B}} p_j x_{j,n+m+1} \tag{5.1a}$$

$$\text{s.t.} \sum_{(i,j)\in\delta(0)} x_{ij} = \kappa, \tag{5.1b}$$

$$\sum_{(i,j)\in\delta(n+m+1)} x_{ij} = \kappa, \tag{5.1c}$$

$$\sum_{(i,j)\in\delta(i)} x_{ij} = 2 \qquad \forall i \in \mathcal{L}, \tag{5.1d}$$

$$\sum_{(i,j)\in\delta(i)} x_{ij} = 2x_{i,n+m+1} \qquad \forall i \in \mathcal{B}, \tag{5.1e}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq \kappa \qquad \forall S \subseteq \mathcal{V},\ 0 \in S,\ n+m+1 \notin S, \tag{5.1f}$$

$$\sum_{j\in\mathcal{B}} p_j x_{j,n+m+1} \geq \beta, \tag{5.1g}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq 2r(S) \qquad \forall S \subseteq \mathcal{L},\ |S| \geq 2, \tag{5.1h}$$

$$x_{ij} \in \{0,1,2,\ldots,K\} \quad \forall (i,j) \in \delta(\mathcal{B}, n+m+1), \tag{5.1i}$$

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{E} \setminus \delta(\mathcal{B}, n+m+1), \tag{5.1j}$$

$$\kappa \in \{1,\ldots,K\}. \tag{5.1k}$$

Equation (5.1a) is the objective function, which aims at minimizing the total costs, i.e., total travel costs minus total revenues obtained from visiting backhaul customers. Equations (5.1b) and (5.1c) impose that exactly $\kappa$ paths are connected to the origin depot 0 and to the destination depot $n+m+1$. Equation (5.1d) ensures that each linehaul customer is visited exactly once by a vehicle by forcing the connection of exactly two edges – one input edge and one output edge. By the definition of the problem any route is allowed to visit a backhaul customer at most once, and a backhaul can be visited by several distinct routes. In this way, Equation (5.1e) makes the number of routes that enters in each backhaul equals the number of routes that connects that backhaul to the destination depot $n+m+1$, taking advantage that there are no edges among the backhauls neither between them and the origin depot. Equation (5.1f) ensures the existence of $\kappa$ routes that begin at the origin depot 0 and end at the destination depot $n+m+1$, accomplishing this by delimiting the number of selected edges to the routes in each possible cut $S$ separating 0 and $n+m+1$. We anticipate that constraints (5.1f) will be dynamically added with the usage of the so-called $\kappa$-cuts. The equations (5.1e) and (5.1f) together guarantee that whether a backhaul is chosen, there is only one in a route and it is connected to the destination depot, i.e., both equations are necessary to assure the precedence constraints among the linehauls and backhauls. Furthermore, these equations assure that there are no routes with two extreme points in 0 nor two extreme points in $n+m+1$.

Equation (5.1g) ensures that a minimum amount of revenue $\beta$ is achieved with a given routing plan. We emphasize that, in practice, the wood industry is concerned with the different characteristics of raw-materials, which have great impact on the manufacturing processes. The work of Andersson et al. [6] distinguishes between "hard" constraints that reflect the logs requirements in the wood industry, and the "soft" constraints that reflect desirable logs properties. The authors also note that both type of constraints are often uncertain.

Equation (5.1h) denotes, simultaneously, the well-known capacity inequalities and the subtour elimination constraints for the part of the routes made only by linehaul customers. The value of $r(S)$ represents the minimum number of routes (vehicles) necessary to attend the demand of the linehaul customers in $S \subseteq \mathcal{L}$. The optimal value of $r(S)$ can be calculated by the Bin Packing Problem [128], which is NP-hard. A widely used lower

bound to $r(S)$ is given by $k(S) = \lceil q(S)/C \rceil$, where $q(S) = \sum_{i \in S} q_i$. These constraints work as follows: *(i)* if $S$ has demand $q(S) > C$, at least $r(S)$ routes must connect $S$ with $\mathcal{V} \setminus S$, and, since there is a factor of 2 on the right hand side, this ensures that a vehicle must enter and exit the cut $S$; and *(ii)* given a route of vertices in $S$ that is a subtour, it is known that connections between $S$ and $\mathcal{V} \setminus S$ are such that $\sum_{(i,j) \in \delta(S)} x_{ij} < 2r(S)$, thus eliminating this route. In this paper, we substitute $r(S)$ by $k(S)$ in Equation (5.1h), giving rise to the well known rounded capacity inequalities (RCIs). We emphasize that this substitution maintains the correctness of the formulation [103]. Equations (5.1i)–(5.1k) define the domain of the decision variables.

**Remark.** If $\beta = 0$ then the deterministic VRPSB can be converted to the Asymmetric Capacitated VRP. To do that, it is necessary to execute the steps described in Algorithm 5.1.

Algorithm 5.1. Elimination of backhauls from the deterministic VRPSB ($\beta = 0$)

1 **for** $j \in \mathcal{L}$ **do**
2 calculate the new traveling costs:

$$c'_{j,n+m+1} = \min \left\{ \min_{i \in \mathcal{B}} \{c_{ji} + c_{i,n+m+1} - p_i\}, c_{j,n+m+1} \right\}$$

3 formulate the model (5.1a)–(5.1k) with $\mathcal{B} = \emptyset$ and new traveling costs $c'_{j,n+m+1} \quad \forall j \in \mathcal{L}$

Algorithm 5.1 calculates new minimal costs for the arcs from the linehaul customers to the depot and then removes the selective backhaul customers from the problem. The algorithm can be performed in $\mathcal{O}(|\mathcal{L}||\mathcal{B}|)$ and its output results in an Asymmetric Capacitated VRP.

### 5.2.2 Robust VRPSB

Robust optimization is a reasonably recent research field [89] that opened a plethora of possibilities to deal with uncertainty in many optimization problems. It is a suitable approach to address uncertainty when its probability distribution is not known, or when infeasibility cannot be tolerated [121, 89]. RO assumes that uncertain parameters belong to a bounded uncertainty set, and the main aim is to provide an optimal solution that is feasible for this entirely uncertainty set [134]. Defining the uncertainty set is, therefore, a major decision that will greatly influence the design of the model and the formulation of the problem. One of the main advantages of RO is precisely its computational tractability for many types of uncertainty sets [89].

In this work, we consider that revenue in backhauls is uncertain. As mentioned in Section 5.1, this is a very common issue in wood-based supply chains, where a minimum amount of revenue related to quantity and quality of raw materials has to be collected everyday and it is not possible to know the quality of wood beforehand. In our approach, revenue uncertainty will only be considered over the Constraint (5.1g), meaning that a minimum revenue requirement of $\beta$ must be satisfied considering the deviations controlled by the uncertainty set. This allows for a trade-off between the nominal profit obtained

in the objective function and the feasibility of the correspondent of Constraint (5.1g) in the robust version of the model. In addition, it will be possible to assign probabilistic bounds of violating the constraint corresponding to the Constraint (5.1g), as it will be detailed in Subsection 5.2.3. Finally, it is noteworthy that this approach is intuitive, meaning that the decision maker aims to optimize the nominal objective function, but at the same time must meet a minimum revenue requirement for all the cases considered in the uncertainty set.

The RO approach used in this work follows the budgeted uncertainty approach of Bertsimas and Sim [23] to develop the robust version of the VRPSB under study. The uncertain revenues are assumed to belong to a space represented by a polyhedral set $\mathcal{U}(\Gamma)$, where $\Gamma$ represents the budget of uncertainty. The choice of the polyhedral set comes with two explanations. First, the resulting model is more computationally tractable than others with non-linear sets, which is desirable for a problem that is already NP-hard. Secondly, by considering a polyhedral set, it is possible to derive tight probabilistic bounds, as shall be further presented in this paper. For more details on the approach, see Bertsimas and Sim [23], Smith and Ahmed [165], Gorissen et al. [89].

The uncertain revenue $\tilde{p}_j$ of each backhaul $j \in \mathcal{B}$ is modeled as a symmetric and independent random variable, bounded by the interval $[\bar{p}_j - \dot{p}_j, \bar{p}_j + \dot{p}_j]$, where $\bar{p}_j \geq 0$ is the nominal revenue of backhaul $j$ and $\dot{p}_j \geq 0$ is the deviation in the revenue. It is thus possible to define a scale deviation $\varepsilon_j$ varying between $[-1, 1]$, such that the revenue of each backhaul is given by

$$\tilde{p}_j = \bar{p}_j + \dot{p}_j \varepsilon_j.$$

Therefore, the scale deviation of each backhauls can be seen as $\varepsilon_j = (\tilde{p}_j - \bar{p}_j)/\dot{p}_j$.

The uncertainty set $\mathcal{U}(\Gamma)$ is represented as follows:

$$\mathcal{U}(\Gamma) = \left\{ \varepsilon \in \mathbb{R}^{|\mathcal{B}|} : -1 \leq \varepsilon_j \leq 1 \quad \forall j \in \mathcal{B}, \sum_{j \in \mathcal{B}} |\varepsilon_j| \leq \Gamma \right\}, \tag{5.2}$$

where $\varepsilon$ is the vector containing the scale deviation $\varepsilon_j$ for each $j \in \mathcal{B}$. As observed from Equation (5.2), the budget of uncertainty $\Gamma$ is introduced in order to control the number of times the worst (most deviated) realizations of revenues are allowed to occur. In the VRPSB, the budget of uncertainty falls in the range $[0, |\mathcal{B}|]$, but $\Gamma = \min\{|\mathcal{B}|, |\mathcal{K}|\}$ is already the worst scenario for the problem, since the maximum number of deviations is limited by either the number of backhauls or the number of vehicles/routes. Therefore, we have that: when $\Gamma = 0$, the model becomes the deterministic version of the problem, i.e., no deviations are considered for any revenue; and when $\Gamma = \min\{|\mathcal{B}|, |\mathcal{K}|\}$, the model becomes its worst case deterministic version, i.e., all revenues suffer deviation, which is equivalent to the approach of Soyster [166]. Thus, $\Gamma$ is used to adjust the robustness of the method against the level of conservatism of the solution [23], or in other words, $\Gamma$ is used to control the size of the uncertainty set.

Based on the uncertainty set $\mathcal{U}(\Gamma)$ and on the deterministic version of the model given by (5.1a)–(5.1k), we now consider that the revenues in Constraint (5.1g) are replaced by

uncertain revenues, giving rise to the following semi-infinite set of constraints:

$$\sum_{j\in\mathcal{B}} \bar{p}_j x_{j,n+m+1} + \sum_{j\in\mathcal{B}} \dot{p}_j x_{j,n+m+1}\varepsilon_j \geq \beta \quad \forall \varepsilon \in \mathcal{U}(\Gamma). \tag{5.3}$$

To carry on the budgeted uncertainty approach [23] to our problem, let us consider a subproblem that searches for a worst case realization of $\varepsilon$ while respecting the budget of uncertainty $\Gamma$. Such a case leads the left hand side of its corresponding inequality in (5.3) to be as small as possible, making valid all the inequalities in (5.3), i.e., for all $\varepsilon \in \mathcal{U}(\Gamma)$. Thus, this subproblem can be denoted by $\min\left\{\sum_{j\in\mathcal{B}} \dot{p}_j x_{j,n+m+1}\varepsilon_j : \varepsilon \in \mathcal{U}(\Gamma)\right\}$, in which the variables are given by $\varepsilon_j$, for all $j \in \mathcal{B}$, and the current values of $x_{j,n+m+1}$, for all $j \in \mathcal{B}$, are considered constant. Regarding an optimal solution to this subproblem, one can observe – taking into account the definition of the uncertainty set $\mathcal{U}(\Gamma)$ – that each variable will have a non-positive value. Therefore, by also considering the structure of $\mathcal{U}(\Gamma)$, we are able to obtain an equivalent subproblem, which appears embedded in the constraint (5.4), that is, by turn, equivalent to the constraints (5.3).

$$\sum_{j\in\mathcal{B}} \bar{p}_j x_{j,n+m+1} - \max_{\varepsilon\in\mathcal{U}(\Gamma)} \sum_{j\in\mathcal{B}} \dot{p}_j x_{j,n+m+1}\varepsilon_j \geq \beta. \tag{5.4}$$

Now we focus on this maximization version of the subproblem. Once more taking into account the structure of $\mathcal{U}(\Gamma)$, the terms $|\varepsilon_j|$, for all $j \in \mathcal{B}$, can be linearized by relying on the fact that each variable will be non-negative in an optimal solution (it is worth noting that a similar reasoning shows us that all variables, except, perhaps, one, will have integer values in such a solution). Therefore, the subproblem can be equivalently formulated as given by equations (5.5a)–(5.5d). For the sake of convenience, we present the correspondent dual variables identifiers on the left of the constraints.

$$\max \quad \sum_{j\in\mathcal{B}} \dot{p}_j x_{j,n+m+1}\varepsilon_j \tag{5.5a}$$

$$(\lambda) \quad \text{s.t.} \quad \sum_{j\in\mathcal{B}} \varepsilon_j \leq \Gamma, \tag{5.5b}$$

$$(\mu) \quad \varepsilon_j \leq 1 \quad \forall j \in \mathcal{B}, \tag{5.5c}$$

$$\varepsilon_j \geq 0 \quad \forall j \in \mathcal{B}. \tag{5.5d}$$

At this point, despite that we were able to move away from a formulation with an infinity number of constraints, we still have the difficulty situation where we have a linear program that contains another linear program embedded in one of its constraints. Let us continue the application of the budgeted uncertainty robust approach, by dealing with the incorporation of the dual of the subproblem into the general problem formulation. Note that the dual variable related to Constraint (5.5b) is given by $\lambda$ and the dual variables related to constraints (5.5c) are given by $\mu_j$, for all $j \in \mathcal{B}$. In the equations (5.6a)–(5.6d)

we present the dual formulation of the subproblem:

$$\min \quad \lambda\Gamma + \sum_{j\in\mathcal{B}} \mu_i \tag{5.6a}$$

$$\text{s.t.} \qquad \lambda + \mu_j \geq \dot{p}_j x_{j,n+m+1} \qquad \forall j \in \mathcal{B}, \tag{5.6b}$$

$$\lambda \geq 0, \tag{5.6c}$$

$$\mu_j \geq 0 \qquad \forall j \in \mathcal{B}. \tag{5.6d}$$

Observe that the subproblem's primal formulation in (5.5a)–(5.5d) is feasible and bounded, and therefore, the strong duality holds. As their optimal objective values coincide, we may substitute the primal subproblem by its dual in the general problem model. In order to obtain a tractable formulation, it is pivotal that we can omit the minimization term of the subproblem's dual formulation when embedding it into the model, as the value of any solution of the dual yields an upper bound to the optimal value of the primal subproblem. Moreover, note that it would not be enough to use a lower bound, which is given by any solution of the primal subproblem.

Finally, following the first theorem of Bertsimas and Sim [23], the dual of the subproblem formulation is incorporated into the model. By this reformulation, we obtain a tractable robust counterpart, as follows:

$$\min \quad \sum_{(i,j)\in\mathcal{E}} c_{ij} x_{ij} - \sum_{j\in\mathcal{B}} \bar{p}_j x_{j,n+m+1} \tag{5.7a}$$

$$\text{s.t.} \quad \text{(5.1b)–(5.1f)},$$
$$\text{(5.1h)–(5.1k)},$$

$$\sum_{j\in\mathcal{B}} \bar{p}_j x_{j,n+m+1} - \left(\lambda\Gamma + \sum_{j\in\mathcal{B}} \mu_j\right) \geq \beta, \tag{5.7b}$$

$$\lambda + \mu_j \geq \dot{p}_j x_{j,n+m+1} \quad \forall j \in \mathcal{B}, \tag{5.7c}$$

$$\lambda \geq 0, \tag{5.7d}$$

$$\mu_j \geq 0 \quad \forall j \in \mathcal{B}. \tag{5.7e}$$

It is worth noting that, in relation to Equation (5.1a), the objective function in (5.7a) was slightly adjusted to only consider the nominal revenues in the robust counterpart. Considering the above reformulation, the robust VRPSB model presents $1 + |\mathcal{B}|$ more variables than the deterministic version, as a result from including the dual variables $\lambda$ and $\mu_j$, for all $i \in \mathcal{B}$. In addition, the robust model also includes $2|\mathcal{B}|+1$ more constraints, as a result of including constraints (5.7c)–(5.7e), with the ones in (5.7d)–(5.7e) being of non-negativity.

Note that the uncertainty only appears in the constraints but not in the objective function. This is aligned with basic assumptions in RO, namely that the objective is certain and the uncertain constraints are hard, in the sense that cannot be violated [89].

As such, the decision-maker is optimizing the expected revenue but concerned that, in the worst case, the minimum revenue required is met. In this work, optimizing over the expected value instead of the worst case value is valid because this operational routing problem is performed every day and worst cases are not usually catastrophic and can be compensated by better scenarios in the next day. Still, we include the constraint of minimum required revenue, because a minimum quality/revenue of raw-material in the worst case scenario may be desirable for the sustainability of the production process.

### 5.2.3 Estimates of Probabilistic Bounds

In RO, it is not necessary to acknowledge the probability distribution of the uncertainty sources. Nevertheless, it is necessary to consider how many revenues will deviate from their nominal value, which is represented by $\Gamma$. The budget of uncertainty determines the size of the uncertainty set and can be defined using historical values and the opinion of experts. $\Gamma$ can also be defined based on probabilistic bounds of constraint violation, as shown in Bertsimas and Sim [23].

In this subsection, we describe how $\Gamma$ can be defined using different probabilistic bounds. We start with an *a priori* probabilistic bound case where no probability distribution is assumed for the uncertainty sources. Then we describe *a priori* and *a posteriori* probabilistic bounds that can be obtained assuming that the uncertainty sources follow an uniform distribution. The last *a priori* one is a method that provides tighter probabilistic bounds for $\mathcal{U}(\Gamma)$ when random parameters are independent, identically and uniformly distributed.

The choice for an uniform distribution comes from the fact that, since the random parameters are not known from historical data, one could assume that all of them can occur with the same probability. It is important to note that it is still a conservative assumption, given that the extreme values have the same probability to occur than the mean value. However, this assumption allows to derive much tighter probabilistic bounds for the problem. Obviously, if it is not possible to assume any distribution, standard probabilistic bounds [23, 93] can still be assumed, as the one described next.

#### *A Priori* Probabilistic Bound Based on Bertsimas and Sim [23]

Based on Bertsimas and Sim [23], the budget of uncertainty can be calculated according to probabilistic bounds of constraint violation without assuming any distribution for the uncertain parameters. Assuming that $\varepsilon_j$, for all $j \in \mathcal{B}$, are independent, symmetrically distributed, and have the expected value of 0, the bound of violating the Constraint (5.7b) can be calculated as follows:

$$\Pr\left(\sum_{j\in\mathcal{B}} \tilde{p}_j x_{j,n+m+1} < \beta\right) \approx 1 - \phi\left(\frac{\Gamma - 1}{\sqrt{|\mathcal{B}|}}\right), \tag{5.8}$$

where $\phi(\cdot)$ is a standard Gaussian cumulative distribution function. According to Equation (5.8), the probability of constraint violation is expected to decrease as the budget of uncertainty increases. Therefore, this bound can be used to estimate the budget of

uncertainty based on confidence levels and avoid solving the problem several times to obtain an adequate budget of uncertainty.

### *A Priori* Probabilistic Bound Based on Guzman et al. [94] and Kang et al. [106]

If the probability distribution function of the uncertainty parameters is assumed to be known, it is possible to derive tighter probabilistic bounds, as presented by Guzman et al. [94] and Kang et al. [106]. Assuming that each uncertain parameter $\varepsilon_j$, for all $j \in \mathcal{B}$, is independent, has the expected value equal to 0, and has a known cumulant generating function given by $\Lambda_j(\theta)$, the following bound holds:

$$\Pr\left(\sum_{j\in\mathcal{B}} \tilde{p}_j x_{j,n+m+1} < \beta\right) \leq \exp\left(\min_{\theta>0}\left\{-\theta\Gamma + \sum_{j\in\mathcal{B}} \Lambda_j(\theta)\right\}\right). \tag{5.9}$$

Here, we assume that each parameter $\varepsilon_j$ follows an uniform distribution supported in $[-1, 1]$, which gives $\Lambda_j(\theta) = \ln\left(\frac{\sinh\theta}{\theta}\right)$. Finally, $\Gamma$ in Equation (5.9) is obtained by means of Algorithm 1 from Guzman et al. [94], given a desired probability of constraint violation.

### *A Priori* Probabilistic Bound Based on the Irwin-Hall Distribution of Santos et al. [159]

Now we describe a probabilistic bound assuming that each uncertain parameter $\varepsilon_j$, for all $j \in \mathcal{B}$, is independent and uniformly distributed in $[-1, 1]$. Proposed for this problem by Santos et al. [159], the complete version of this paper, it is inspired by the geometrical interpretation of the uncertainty set $\mathcal{U}(\Gamma)$ through the Irwin-Hall distribution. For further details on the Irwin-Hall distribution, see Marengo et al. [127]. Denoted by $\Pr_+(\Gamma)$, the percentage of realizations considered by $\mathcal{U}(\Gamma)$ can be interpreted as $\Pr\left(\sum_{k=1}^{|\mathcal{B}|} U[-1, 1] \leq \Gamma\right)$, where $U[-1, 1]$ is an uniform distribution in the interval. Note that when $\Gamma$ assumes 0 and $|\mathcal{B}|$, the value of $\Pr_+(\Gamma)$ is 50% and 100%, respectively. In this way, Constraint (5.7b) is satisfied at least with probability $\Pr_+(\Gamma)$ and therefore:

$$\Pr\left(\sum_{j\in\mathcal{B}} \tilde{p}_j x_{j,n+m+1} < \beta\right) \leq 1 - \Pr_+(\Gamma), \tag{5.10}$$

when each $\tilde{p}_j$ is uniformly distributed in $[\bar{p}_j - \dot{p}_j, \bar{p}_j + \dot{p}_j]$.

The computation of $\Pr_+(\Gamma)$ is resorted to the Irwin-Hall distribution, which is the distribution of the sum of $n$ independent and identically distributed uniform random variables in $[0, 1]$. The cumulative distribution of the Irwin-Hall is defined as:

$$F_{\text{IH}}(t, n) = \frac{1}{n!}\sum_{j=0}^{\lfloor t\rfloor}(-1)^j \binom{n}{j}(t-j)^n,$$

where $t \in [0, n]$. The definition of $\Pr_+(\Gamma)$ is given by $F_{\text{IH}}(t, n)$ parameterized to acknowledge the sum of uniform distributions supported in $[-1, 1]$ instead of $[0, 1]$, which is done by $t = (\Gamma + |\mathcal{B}|)/2$ and $n = |\mathcal{B}|$.

Moreover, considering that the maximum number of visited backhauls is limited by $\min\{|\mathcal{B}|, |\mathcal{K}|\}$, a tighter probabilistic bound can be achieved. Hence, $\Pr_+(\Gamma)$ can be tightened with the usage of $t = (\Gamma + \min\{|\mathcal{B}|, |\mathcal{K}|\})/2$ and $n = \min\{|\mathcal{B}|, |\mathcal{K}|\}$ in $F_{\text{IH}}(t, n)$, which is denoted by $\Pr^*_+(\Gamma)$. Therefore, analogous to Equation (5.10), a tighter probabilistic bound can be obtained as follows:

$$\Pr^* \left( \sum_{j \in \mathcal{B}} \tilde{p}_j x_{j,n+m+1} < \beta \right) \leq 1 - \Pr^*_+(\Gamma). \tag{5.11}$$

#### *A Posteriori* Probabilistic Bound Based on Guzman et al. [95]

All the above methods to obtain probabilistic bounds are referred to as *a priori* bounds, which means that the probabilistic bound of violating a constraint is determined before considering the solution of the optimization model. The probabilistic bound GMF12 in Guzman et al. [95] estimates an *a posteriori* probability of constraint violation. Although this method is more challenging than the previous ones, it provides a tighter bound because it allows to characterize one single solution to the problem [95], rather than every feasible solution of the uncertainty set. Considering that each $\varepsilon_j$, for all $j \in \mathcal{B}$, is independent and follows a continuous uniform distribution supported in $[-1, 1]$, the *a posteriori* (i.e., exact) probability of constraint violation of a solution $x^*$ can be defined by the following equation:

$$\Pr \left( \sum_{j \in \mathcal{B}} \bar{p}_j x^*_{j,n+m+1} + \sum_{j \in \mathcal{B}^*} \dot{p}_j x^*_{j,n+m+1} \varepsilon_j < \beta \right) = F_\Theta \left( h(x^*) \right), \tag{5.12}$$

where $F_\Theta(\cdot)$ is the cumulative distribution function of $\sum_{j \in \mathcal{B}} \dot{p}_j x^*_{j,n+m+1} \varepsilon_j$, the deterministic part of the constraint is given by $h(x^*) = \beta - \sum_{j \in \mathcal{B}} \bar{p}_j x^*_{j,n+m+1}$, and $\mathcal{B}^* = \{j : \dot{p}_j x^*_{j,n+m+1} > 0\}$. Because calculating $F_\Theta(\cdot)$ is quite challenging, the brute-force enumeration algorithm proposed by Guzman et al. [95] is used to calculate it.

## 5.3 Solution Methods

This section presents an exact method, that is a BC algorithm, and an ALNS metaheuristic to solve the robust VRPSB. We devised our BC algorithm following the work of Lysgaard et al. [125], that is combined with the standard MILP solver CPLEX[1]. The ALNS algorithm was proposed to this problem by Santos et al. [159], the complete version of this paper, and it is based on the work of Ropke and Pisinger [154] with the inclusion of procedures to incorporate the robust nature of the problem. We anticipate that both of the solution methods presented are further used to solve small, medium and large instances adapted from literature, including their computational performance comparison.

[1] https://www.ibm.com/products/ilog-cplex-optimization-studio

### 5.3.1 Branch-and-Cut

The BC algorithm developed is partially based on the BC for the Symmetric Capacitated VRP (SymCVRP) presented in Lysgaard et al. [125], which is well known for its efficiency. Since that work, the methods for the SymCVRP itself evolved and the state-of-the-art algorithms heavy rely on the branch-cut-and-price (BCP) framework [146], which involves separation algorithms to identify promising cuts and pricing algorithms to generate suitable columns for the problem, as these models are based on set partitioning-like formulations [170]. Our exact approach is based on BC, which can be viewed as a component to an eventual BCP based method.

The authors of Lysgaard et al. [125] have made the source code of a software library called CVRPSEP publicly available, which implements several cut separation routines for the SymCVRP. In this paper, some of these separation routines are used in our implementation of the BC to solve the robust VRPSB. More precisely, our BC algorithm makes use of the RCI separation routine from the CVRPSEP, as well as it utilizes the max-flow routine, from the same library, to separate the $\kappa$-cuts we used in our approach. For a comprehensive material about the separation algorithms we refer to Toth and Vigo [170]. The BC is applied to the resolution of the robust VRPSB formulation, which is presented in Subsection 5.2.2, given by the following equations: (5.7a), (5.1b)–(5.1f), (5.1h)–(5.1k), and (5.7b)–(5.7e).

The formulation presents suitable features for the BC method we developed, namely: *(i)* the undirected graph, *(ii)* the split depot, and *(iii)* the $\kappa$-cuts. Using an undirected graph to represent the network in the VRPSB avoids symmetric solutions and keeps the number of variables in the formulation smaller than a hypothetically directed graph. In the VRPSB, it is mandatory to constrain a backhaul vertex to be, at most, on one extreme point of a route, and therefore, the identification of the depot-connected edges of a given route is essential. To accomplish that we splitted up the depot into the origin depot 0 and the destination depot $n+m+1$, as already mentioned. Considering this transformation, the problem can be seen as seeking paths that connects 0 to $n+m+1$, and, if a backhaul customer is present in a path, it must be connected to the $n+m+1$. A linehaul customer can also be connected to $n+m+1$, meaning that its route has no backhaul at all. On the other hand, we must ensure that there are no routes where there are two extreme points in 0 nor two extreme points in $n+m+1$. The $\kappa$-cuts in Equations (5.1f) solve this problem, as those cuts restrict the solution to have at least $\kappa$ paths between 0 and $n+m+1$ in a solution.

As there is an exponential number of $\kappa$-cuts, we rely on the BC capabilities for seeking and adding violated cuts to the model. To separate these constraints, we use the implementation of the max-flow algorithm, presented in the CVRPSEP. If the max-flow value is smaller than $\kappa$ that cut is violated and added to the model. Note that Equations (5.1b) and (5.1c) are specialized and stronger $\kappa$-cuts.

The RCIs, at Equation (5.1h), ensures that the capacity limit of each vehicle is respected. Each time the BC has to call the RCI separation routine from the Lysgaard's CVRPSEP, it is necessary to make a conversion on the graph. This transformation occurs as follows: given a candidate solution, the vertices 0, $n+m+1$, and the vertices of $\mathcal{B}$

are all shrunk into a single supervertex that plays the role of the depot on the separation algorithm. If violated cuts are found, and considering that the edges are not incident to the depot, the cuts are added to the model and it is re-optimized.

The BC algorithm adds the valid inequalities described above via the BC framework of CPLEX by using its callback functions. The separation step is composed by four heuristics for the RCIs, accordingly Lysgaard et al. [125], and the above-mentioned max-flow algorithm for the $\kappa$-cuts. The branching strategy is kept as the default branching of CPLEX, which typically branches on variables. An RCI is considered violated if the difference between its left and right hand sides is greater than an RCI tolerance limit. In this work, this limit is 0.2. A $\kappa$-cut is considered violated if the max-flow value is smaller than $\kappa$.

Finally, the model initially given to the BC algorithm is composed by the objective function in Equation (5.7a), the constraints in equations (5.1b)–(5.1e) and (5.7b)–(5.7c), the bounds in equations (5.1i)–(5.1k) and (5.7d)–(5.7e), and two additional cuts in a similar form of Equation (5.1h): one for the origin depot with $S = \{0\}$, and the other for the destination depot with $S = \{n + m + 1\}$, both with the right-hand side as $r(\mathcal{L})$ (which is substituted by $k(\mathcal{L})$ in the resolution). As the model defines an exponential number of $\kappa$-cuts in Equation (5.1f) and also an exponential number of RCI constraints in Equation (5.1h), they are added on demand in a branch-and-cut fashion.

### 5.3.2 Adaptive Large Neighborhood Search

In this subsection we describe the ALNS applied to the resolution of the robust VRPSB proposed by Santos et al. [159] (the complete version of this paper), which, by its turn, is based in the general method of Ropke and Pisinger [154]. The ALNS was proposed by Ropke and Pisinger [154] and since then, it has shown to be effective for many VRPBs [129, 120, 84], as well as for other VRPs [118, 52, 183]. For an overview on the ALNS metaheuristic we refer to Pisinger and Ropke [145].

Large neighborhoods can be explored by consecutively applying two heuristics: first, a *destroy* operator, which partially destroys the current solution, and then a *repair* operator, which make the solution feasible again, thus generating a new solution. The main features of ALNS are the usage of diverse destroy and repair operators to explore different neighborhoods [145] and the adaptive search behavior as a consequence of the performance of each pair of destroy-repair operators during the iterative process.

In the robust ALNS for the robust VRPSB, the destroy operators used are the *Shaw removal*, the *random removal*, and the *worst removal* heuristics. The Shaw removal selects the vertices to remove according to similarity criteria, namely distance and quantity [163]; the random removal selects at random the vertices to remove; and the worst removal selects to remove the nodes that would highly decrease the objective value. The repair operators used are the *greedy insertion* and the *k-regret insertion*. The former heuristic is very simple, inserting the removed nodes into positions in the routes such that the objective function increases the minimum possible. The latter applies the concept of *regret* to insert first the customers for which the insertion in a later stage would be more costly (with higher regret).

In this approach, robustness was embedded into the algorithm during the creation of new neighbors. Thus, instead of creating a new solution with the destroy-repair operators and afterwards evaluating its robustness, the robustness aspects are already evaluated alongside the application of the selected repair operator, as described in the following.

Over a solution after the application of the destroy operator, which may contain some linehauls and backhauls, the repair operator starts by inserting all the missing linehaul customers, obtaining a solution $x'$. After that, the total of nominal (expected) revenues, the total of revenues with deviations, and the robustness of the solution are assessed. For the computation of the total of revenues with deviations, the backhauls in $x'$ are sorted in decreasing order of the values of revenue deviation multiplied by the number of times that the backhaul is present in the solution. Afterwards, the revenues with deviations are iteratively summed through this sequence of backhauls, and simultaneously, each backhaul consumes one unit in the budget of uncertainty $\Gamma$, except, perhaps, the last one before the budget is fully consumed. The last one's deviation is considered proportionally to what remains in $\Gamma$, while the ones after that have only the nominal revenue summed up. At this point, the feasibility of the solution is accessed by checking if the total of revenues with deviations is sufficient to achieve the minimum revenue required $\beta$. If the solution is not feasible, its nominal revenue takes $-\infty$.

Subsequently, the repair operator in use proceeds by inserting potential backhauls in available routes and creates a new solution $x''$, that has its attributes evaluated as described before. Finally, the solution with the lowest total cost (total distance minus total nominal revenue) between $x'$ and $x''$ is selected as the next neighbor solution in the ALNS algorithm.

## 5.4 Computational Tests

The aim of the computational tests is twofold: *(i)* evaluate the performance of models and robustness of solutions, namely in terms of probabilistic bounds, feasibility, and robustness; and *(ii)* evaluate the performance of the BC to solve the robust VRPSB and compare it to the ALNS.

The first aim is addressed with experiments over a few selected small/medium instances with a single-commodity-flow formulation of the problem. Instead of the two-index vehicle-flow formulation presented in Section 5.2, the commodity-flow model was used for this, as it is a simpler formulation, avoids the exponential constraints in equations (5.1f) and (5.1h), and has low resolution times to exactly solve the selected instances. In fact, the resolution time is not a concern at this point, since the main goal is to analyze the structure of the solutions and their feasibility for different robustness parameters. In this aim's context, this formulation is implemented in the OPL modeling language and solved using the MILP solver CPLEX 12.6.

The second aim is accomplished by a series of experiments on a set of instances including medium and large ones. The BC algorithm, over the vehicle-flow formulation presented in Section 5.2, is implemented in C++, using the C callable API of CPLEX 12.7.1. The experiments with the BC were run on a computer with the processor Intel

Xeon Gold 6142 of 2.60 GHz with 64 physical cores, 256 GB of RAM, and the GNU/Linux Ubuntu 18.04.2 operating system. Each BC test was performed sequentially, although at most 32 of them were in parallel at any time in the machine. The ALNS algorithm is implemented in C++. The experiments with the ALNS were performed on a computer equipped with the processor Intel Core i7 of 2.20 GHz, 16 GB of RAM, and the Windows 10 operating system.

### 5.4.1 Problem Instances

The instances used in the experiments are adapted from the standard VRPB instances in Goetschalckx and Jacobs-Blecha [87]. The new parameters added to these instances are the revenues and deviations, which are randomly generated for each backhaul of each instance as follows. From the original class of instances B of Goetschalckx and Jacobs-Blecha [87], we have used their quantities as proxies for the minimum and maximum revenues. Then, 50 values are randomly simulated within this range, for each backhaul. The average value from this simulation corresponds to the revenue, $\bar{p}$, and the standard deviation corresponds to the deviation, $\dot{p}$. Furthermore, these values are simulated for the 10 backhaul customers from class B instances and then replicated for instances with higher values, such that the revenue and corresponding deviation for backhaul customer 1 are the same for backhaul customers 11, 21, and so on. These values are reported in Table 5.1.

Table 5.1. Values of revenue $\bar{p}$ and deviation $\dot{p}$ for each backhaul customer used for instances in the computational tests

| Backhaul number | $\bar{p}$ | $\dot{p}$ |
|---|---|---|
| 1, 11, 21, 31, 41, 51, 61, 71 | 438 | 242 |
| 2, 12, 22, 32, 42, 52, 62, 72 | 404 | 194 |
| 3, 13, 23, 33, 43, 53, 63, 73 | 454 | 239 |
| 4, 14, 24, 34, 44, 54, 64, 74 | 392 | 231 |
| 5, 15, 25, 35, 45, 55, 65, 75 | 464 | 229 |
| 6, 16, 26, 36, 46, 56, 66 | 380 | 201 |
| 7, 17, 27, 37, 47, 57, 67 | 384 | 235 |
| 8, 18, 28, 38, 48, 58, 68 | 355 | 222 |
| 9, 19, 29, 39, 49, 59, 69 | 459 | 229 |
| 10, 20, 30, 40, 50, 60, 70 | 415 | 253 |

### 5.4.2 Evaluation of the VRPSB Under Uncertainty

Throughout this subsection, several tests are conducted in order to analyze and evaluate the RO model for the VRPSB presented in Subsection 5.2.2. First, the different probability bounds obtained by the different methods that estimate *a priori* the probability of constraint violation are characterized. Next, the *a posteriori* probability of constraint violation of solutions are presented and compared with probabilistic bounds calculated by an *a priori* method. Then, the nominal and robust solutions obtained for different

values of $\beta$ and $\Gamma$ are analyzed. The structure of nominal and robust solutions are finally analyzed and discussed.

The tests presented in this subsection were performed with four instances and four different values of minimum revenue required to cover a wide range of instance characteristics. As noted before, they were solved by a commodity-flow formulation. The instances tested are: A4 (20 linehaul customers, 5 backhauls, and 3 vehicles), B3 (20 linehaul customers, 10 backhauls, and 3 vehicles), C4 (20 linehaul customers, 20 backhauls, and 4 vehicles) and F4 (30 linehaul customers, 30 backhauls, and 4 vehicles). The values of minimum revenue required, given by $\beta$, are 0, 500, 750, and 1000.

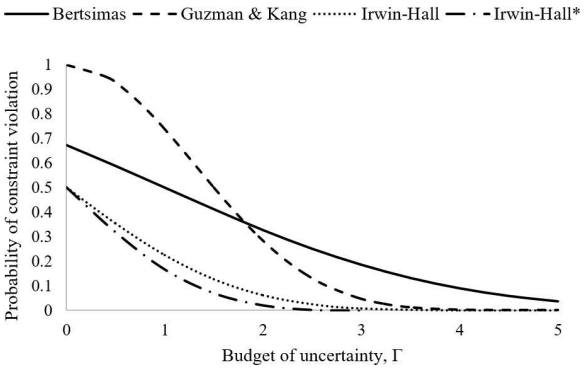
### A Priori Probabilistic Bounds

Here, the methods that estimate the probability of violating Constraint (5.7b), based on the values of the budget of uncertainty $\Gamma$, are analyzed and compared. The calculations for the instances with $|\mathcal{B}| = 5$ and $|\mathcal{K}| = 3$ (A4), $|\mathcal{B}| = 10$ and $|\mathcal{K}| = 3$ (B3), $|\mathcal{B}| = 20$ and $|\mathcal{K}| = 4$ (C4), and $|\mathcal{B}| = 30$ and $|\mathcal{K}| = 4$ (F4) are presented in Figure 5.2. In the figure (and text), the methods of probabilistic bounds denoted by "Bertsimas" and "Guzman & Kang", refer to the ones based on, respectively, Bertsimas and Sim [23] given in Equation (5.8), and Guzman et al. [94] and Kang et al. [106] in Equation (5.9). The other methods are the ones based on the Irwin-Hall distribution devised by Santos et al. [159], where "Irwin-Hall" refers to the one in Equation (5.10) and "Irwin-Hall*" refers to the more restricted one in Equation (5.11). It is important to note that the *a posteriori* method based on Guzman et al. [95] is not analyzed here, since it is not influenced by the robust parameter.

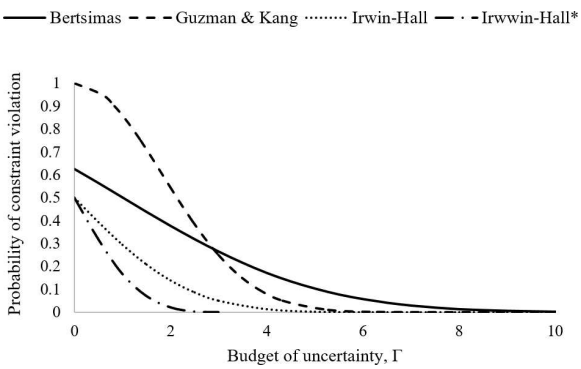
From the calculated results in Figure 5.2, as expected, the probability of constraint violation decreases as the budget of uncertainty increases, for all the methods investigated. Also, as more backhauls exist in an instance, the sooner the probabilistic bound reaches its minimum, i.e., the sharper is the curve provided by the bound.

The method based on Bertsimas and Sim [23] is the most conservative among the four *a priori* probabilistic bounds (except for small values of $\Gamma$ of another method). In fact, when the instance only has 5 backhauls (Figure 5.2a), the probability of constraint violation never reaches zero. When the number of backhauls increases to 10 (Figure 5.2b), a probability of zero only occurs when $\Gamma$ is maximum, i.e., when $\Gamma = |\mathcal{B}|$. In the other cases (figures 5.2c and 5.2d), a probability of zero is achieved but only latter than any of the other methods. The method based on Guzman et al. [94] and Kang et al. [106], by its turn, is very conservative for the lowest values of $\Gamma$ in each instance, but the probability of constraint violation decreases drastically with the increase of the degree of conservatism.

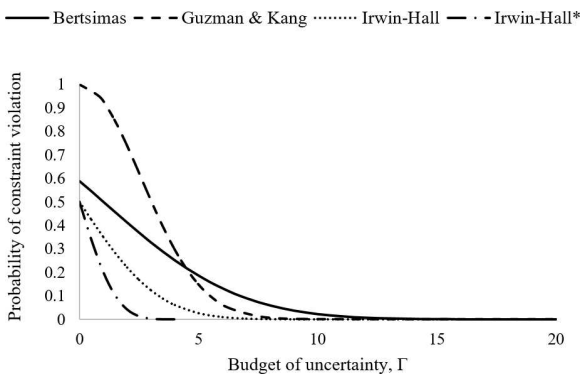
Finally, the methods based on the Irwin-Hall distribution [159] are indisputably the ones that provide the tighter bounds, showing always the lowest values for the probability of constraint violation for any value of $\Gamma$. In fact, by considering that the worst case occurs when $\Gamma_{\max} = \min\{|\mathcal{B}|, |\mathcal{K}|\}$ (i.e., using Irwin-Hall*), it is possible to obtain an even tighter bound than when $\Gamma_{\max} = |\mathcal{B}|$ (i.e., using Irwin-Hall). It is important to note that when the Irwin-Hall* method is used, the same probabilistic bounds are achieved for the cases illustrated in figures 5.2a and 5.2b, since on both cases the number of vehicles



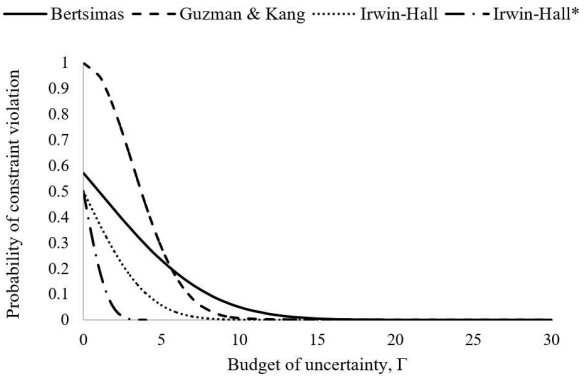

(a) Instance with $|\mathcal{B}| = 5$ and $|\mathcal{K}| = 3$ (A4)




(b) Instance with $|\mathcal{B}| = 10$ and $|\mathcal{K}| = 3$ (B3)




(c) Instance $|\mathcal{B}| = 20$ and $|\mathcal{K}| = 4$ (C4)




(d) Instance $|\mathcal{B}| = 30$ and $|\mathcal{K}| = 4$ (F4)

Figure 5.2. Probability of constraint violation obtained with *a priori* methods over selected instances

is three, which is lower than the number of backhauls. Similarly, the same bounds are achieved for the cases illustrated in figures 5.2c and 5.2d, where the number of vehicles is four.

### Comparison of *a Priori* and *a Posteriori* Probabilistic Bounds

Now we focus on the method that estimates *a posteriori* the probability of constraint violation, i.e., the one based on Guzman et al. [95], which is compared with the best *a priori* method, as concluded above, i.e., the Irwin-Hall* method. The former method may be denoted as "Guzman *post*".

Tables 5.2 and 5.3 report the different values of probability of constraint violation for several values of the budget of uncertainty $\Gamma$ over instances with three and four vehicles, respectively. The first columns of the tables indicate "instance", "$\beta$", and "method" (of probabilistic bound). For the results on Guzman *post*, all the solutions considered for a reported probability are optimal, while a symbol "–" indicates that the instance was concluded to be infeasible with the related value of $\Gamma$. Since the method denoted by Irwin-Hall* only take into account the value of budget of uncertainty $\Gamma$ and the value given by $\min\{|\mathcal{B}|, |\mathcal{K}|\}$, the corresponding probability values are only presented once for each instance without a corresponding value of $\beta$. On the other hand, the method based on Guzman et al. [95] depends on the value of $\beta$ (as well as on the specific backhaul

customers visited and their respective number of visits), thus the tables report the data for several values of $\beta$ for each instance.

Table 5.2. Probability of constraint violation obtained with *a priori* and *a posteriori* methods for two instances with three vehicles

| | | | Budget of uncertainty ($\Gamma$) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Instance | $\beta$ | Method | 0 | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 |
| A4 | | Irwin-Hall* | 50.00% | 31.77% | 16.67% | 7.03% | 2.08% | 0.26% | 0.00% |
| | 0 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 500 | Guzman *post* | 0.85% | 0.85% | 0.85% | 0.85% | 0.00% | 0.00% | 0.00% |
| | 750 | Guzman *post* | 26.17% | 26.17% | 0.69% | 0.69% | 0.25% | 0.14% | – |
| | 1000 | Guzman *post* | 14.41% | 14.41% | 8.97% | – | – | – | – |
| B3 | | Irwin-Hall* | 50.00% | 31.77% | 16.67% | 7.03% | 2.08% | 0.26% | 0.00% |
| | 0 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 500 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 750 | Guzman *post* | 2.21% | 2.21% | 2.21% | 1.42% | 1.42% | 0.20% | – |
| | 1000 | Guzman *post* | 16.69% | 16.69% | 15.38% | 6.40% | – | – | – |

Table 5.3. Probability of constraint violation obtained with *a priori* and *a posteriori* methods for two instances with four vehicles

| | | | Budget of uncertainty ($\Gamma$) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | $\beta$ | Method | 0 | 0.5 | 1 | 1.5 | 2 | 2.5 | 3 | 3.5 | 4 |
| C4 | | Irwin-Hall* | 50.00% | 33.81% | 20.05% | 10.11% | 4.17% | 1.32% | 0.26% | 0.02% | 0.00% |
| | 0 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 500 | Guzman *post* | 1.07% | 1.07% | 1.07% | 1.07% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 750 | Guzman *post* | 22.95% | 22.95% | 0.29% | 0.29% | 0.29% | 0.12% | 0.00% | 0.00% | 0.00% |
| | 1000 | Guzman *post* | 8.15% | 8.15% | 8.15% | 6.40% | 0.23% | 0.34% | 0.17% | 0.01% | – |
| F4 | | Irwin-Hall* | 50.00% | 33.81% | 20.05% | 10.11% | 4.17% | 1.32% | 0.26% | 0.02% | 0.00% |
| | 0 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 500 | Guzman *post* | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% | 0.00% |
| | 750 | Guzman *post* | 0.38% | 0.38% | 0.38% | 0.38% | 0.38% | 0.12% | 0.00% | 0.00% | 0.00% |
| | 1000 | Guzman *post* | 10.07% | 10.07% | 10.07% | 6.26% | 0.18% | 0.18% | 0.18% | 0.02% | – |

The results from both tables 5.2 and 5.3 allow to conclude that much tighter bounds can be derived for any instance, for any $\beta$ and for any $\Gamma$ using the method that estimates probabilistic bounds *a posteriori*. An interesting observation is that it seems that the method based on Guzman et al. [95] estimates a lower probabilistic bound for solutions that visit the same backhaul more than once. Taking for example instance C4 with $\beta = 1000$, where it is shown that the probability of constraint violation when $\Gamma = 2$ is lower than when $\Gamma = 2.5$: in the former case, the backhaul customer $n + 15$ is visited twice while in the later case, it is only visited once.

### The Price of Robustness

Here we analyze the RO model under uncertainty. The performance metric used is the Price of Robustness (PoR) [23, 89], which measures the trade-off between the probability of constraint violation and the effect on the nominal objective value. The PoR is determined as the difference between robust and nominal solutions' values.

The results are reported for instances with three and four vehicles respectively in tables 5.4 and 5.5. The first columns of the tables inform "instance", "$\beta$", and "nominal solution" (that contains the objective value of it). In the sequence, there are the columns containing the PoR for several levels of probability of constraint violation. Besides, all the solutions considered for a reported value are optimal, while a symbol "–" indicates that the instance was concluded to be infeasible with the related value of $\Gamma$. The probability bounds used are the ones given in tables 5.2 and 5.3 for the Irwin-Hall* method on the corresponding instances. Thus, a probability of 50% corresponds to $\Gamma = 0$, a probability of 0% corresponds to the maximum $\Gamma$ applied, and so forth.

Table 5.4. PoR obtained for RO solutions of instances with three vehicles

| | | | Probability of constraint violation (and budget of uncertainty) | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Instance | $\beta$ | Nominal solution | 50% ($\Gamma = 0$) | 31.77% ($\Gamma = 0.5$) | 16.67% ($\Gamma = 1$) | 7.03% ($\Gamma = 1.5$) | 2.08% ($\Gamma = 2$) | 0.26% ($\Gamma = 2.5$) | 0% ($\Gamma = 3$) |
| A4 | 0 | 136,068 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 500 | 136,581 | 0 | 0 | 0 | 0 | 3,193 | 3,193 | 3,193 |
| | 750 | 136,581 | 0 | 0 | 3,193 | 3,193 | 4,284 | 6,719 | – |
| | 1000 | 139,774 | 0 | 0 | 1,091 | – | – | – | – |
| B3 | 0 | 132,731 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 500 | 132,731 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 750 | 132,731 | 0 | 0 | 0 | 770 | 770 | 1,623 | – |
| | 1000 | 132,731 | 0 | 0 | 770 | 9,635 | – | – | – |

Table 5.5. PoR obtained for RO solutions of instances with four vehicles

| | | | Probability of constraint violation (and budget of uncertainty) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Instance | $\beta$ | Nominal solution | 50% ($\Gamma = 0$) | 33.81% ($\Gamma = 0.5$) | 20.05% ($\Gamma = 1$) | 10.11% ($\Gamma = 1.5$) | 4.17% ($\Gamma = 2$) | 1.32% ($\Gamma = 2.5$) | 0.26% ($\Gamma = 3$) | 0.02% ($\Gamma = 3.5$) | 0% ($\Gamma = 4$) |
| C4 | 0 | 128,792 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 500 | 131,464 | 0 | 0 | 0 | 0 | 2,117 | 2,117 | 2,117 | 2,117 | 2,117 |
| | 750 | 131,464 | 0 | 0 | 2,117 | 2,117 | 2,117 | 5,371 | 11,245 | 11,245 | 11,245 |
| | 1000 | 133,580 | 0 | 0 | 0 | 3,254 | 9,341 | 9,932 | 10,035 | 15,317 | – |
| F4 | 0 | 145,456 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 500 | 145,456 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 750 | 145,456 | 0 | 0 | 0 | 0 | 0 | 1,861 | 9,254 | 9,254 | 9,254 |
| | 1000 | 145,456 | 0 | 0 | 0 | 1,861 | 9,253 | 9,254 | 9,254 | 15,061 | – |

First, the analysis of both tables indicate that, as expected, the PoR increases while decreasing the probability of constraint violation (or increasing the budget of uncertainty $\Gamma$), for a given level of $\beta$. When the PoR remains the same, it means that the solution structure also remains the same, and so it does not impact the total expected revenue. These values provide the minimum value of budget of uncertainty that defines the worst

case for a given $\beta$, for each instance. Secondly, increasing the value of $\beta$ tends to force the creation of a new routing plan, probably due to the need of augmenting the visits to backhaul customers. Taking for example, the results obtained with instance B3: when $\beta = 500$, the PoR $= 0$ and remains the same for all levels of probability; when $\beta = 750$, the PoR changes when the probability is 7.03%; and when $\beta = 1000$, the PoR changes sooner, when the probability is 16.67%. Nevertheless, these results are not always in accordance, as the tests with instance C4 reveals that when $\beta = 750$, the PoR changes sooner than when $\beta = 1000$, although the PoR value is higher for the highest $\beta$.

#### Solution Structure

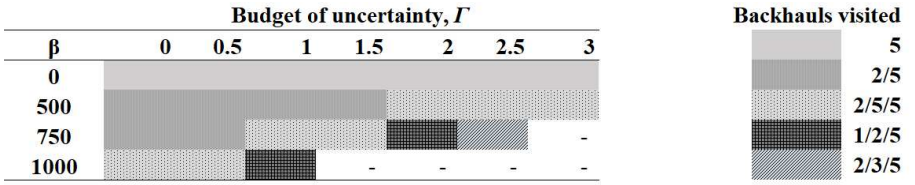
For each value of $\beta$ and $\Gamma$, the solution structure (routing plan) obtained with each instance tested with the RO model is presented in Figure 5.3, specifying which backhaul customers are visited in each routing plan. All solutions depicted in figures 5.3a and 5.3b comprise a plan with three routes, while all the solutions in figures 5.3c and 5.3d comprise a plan with four routes.

As expected, increasing the minimum revenue required leads to increase the number of backhaul customers visited, except in the case of instance B3 (Figure 5.3b). Although, in this case, the first routing plan obtained includes two visits to the same backhaul customer $n + 1$, whereas all the other plans include distinct backhaul customers. It also seems that, at least, one specific backhaul is always included in any route. For example, backhaul $n + 5$ is included in any routing plan of instance A4, backhaul $n + 1$ in instance B3, backhaul $n + 15$ in instance C4 and backhaul $n + 9$ in instance F4.

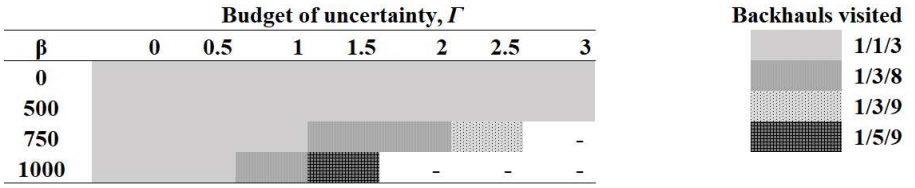
With the exception of the instance C4, at least one backhaul customer is visited when there is no requirement of minimum revenue, i.e., when $\beta = 0$. This means that in these cases, it is profitable to visit a backhaul customer, even if there is no need to. Also, in instances A4 and C4, all routing plans obtained when there is a minimum revenue required are, indeed, different than the cases when such requirement does not exist. In fact, it seems that both instances are very sensitive to changes in any parameter, $\beta$ or $\Gamma$. On the other hand, for instances B3 and F4, the same routing plan is obtained whenever $\beta = 0$ or $\beta = 500$ or even for any value of $\beta$ with, at least, a $\Gamma \leq 0.5$.

### 5.4.3 Evaluation of Solution Methods

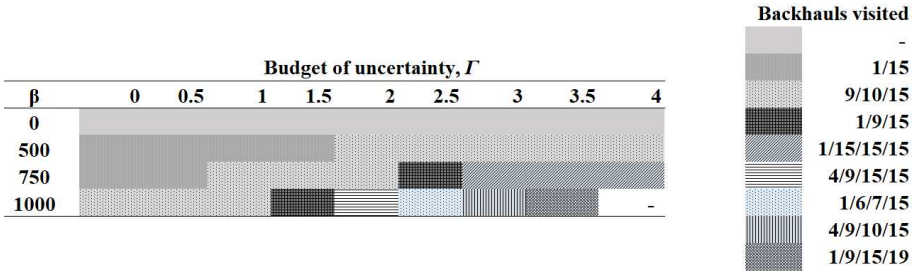
To evaluate the computational performance of the BC and compare it to the ALNS, both methods were tested for instances with different vehicle capacities $C$, numbers of vehicles $|\mathcal{K}|$, linehaul customers $|\mathcal{L}|$ and backhaul customers $|\mathcal{B}|$. These instances are randomly selected from the universe of Goetschalckx and Jacobs-Blecha [87], such that, at least, two instances from each class (from A to N) are obtained (see Table 5.6). The instances are divided into 14 classes, labeled by the letters A–N. The classes G and L have, each one, 3 instances, and the remaining classes have 2 instances each. The instances of a class are almost equal: they are different only in respect to the vehicle capacity $C$ and the number of vehicles $|\mathcal{K}|$ while their graphs, costs, revenues, and deviations are identical. The number of backhaul customers $|\mathcal{B}|$, in relation to $|\mathcal{L}|$, corresponds to: 25% in class A; 50% in B; 100% in C; about 25% in D; 50% in E; 100% in F; about 25% in G; about



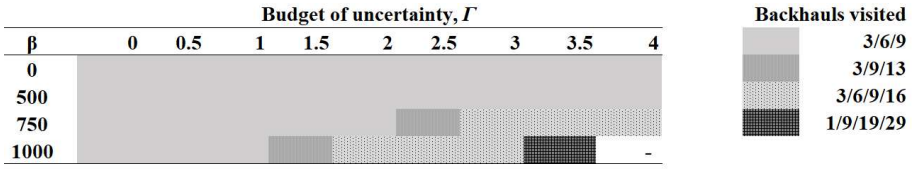

(a) Backhauls on solutions for the instance A4 (three routes)




(b) Backhauls on solutions for the instance B3 (three routes)




(c) Backhauls on solutions for the instance C4 (four routes)




(d) Backhauls on solutions for the instance F4 (four routes)

Figure 5.3. Solution structures of the RO model obtained by changing $\beta$ and $\Gamma$ (backhaul numbers with $n$ subtracted)

50% in H; 100% in I; about 25% in J; about 50% in K; 100% in L; 25% in N; and 50% in M.

Each instance is tested for two different values of budget of uncertainty, $\Gamma = 0$ and $\Gamma = 2$, and for two different values of minimum revenue required, $\beta = 0$ and $\beta = 750$. For the BC, the running time is limited to one hour for each test and is worth recovering that the RCI tolerance is 0.2. For the ALNS [159], 30 replications of each test were performed, each one with the stopping criterion of 10,000 iterations.

The results obtained from these tests are presented in three groups of columns in Table 5.7. The first group presents the instance name ("Inst"), the budget of uncertainty $\Gamma$, and the minimum revenue required $\beta$. The next group refers to the results obtained from the BC, given by the lower bound (LB), the solution objective value ($z$), the gap of the BC ($G_{BC}$), and the time of the test in seconds (T(s)). The $G_{BC}$ of a solution corresponds to the percentage difference between the solution objective value of the BC and its respective LB. This gap is calculated by $G_{BC} = ((z - LB)/LB)100$. We point out that the values of LB and $z$ presented and used to calculate the gaps were rounded

Table 5.6. Instance characteristics

| Inst | $C$ | $\lvert\mathcal{K}\rvert$ | Inst | $C$ | $\lvert\mathcal{K}\rvert$ | Inst | $C$ | $\lvert\mathcal{K}\rvert$ | $\lvert\mathcal{L}\rvert$ | $\lvert\mathcal{B}\rvert$ |
|---|---|---|---|---|---|---|---|---|---|---|
| A2 | 2550 | 5 | A4 | 4050 | 3 | | | | 20 | 5 |
| B1 | 1600 | 7 | B2 | 2600 | 5 | | | | 20 | 10 |
| C1 | 1800 | 7 | C4 | 4150 | 4 | | | | 20 | 20 |
| D2 | 1700 | 11 | D3 | 2750 | 7 | | | | 30 | 8 |
| E1 | 2650 | 7 | E3 | 5225 | 4 | | | | 30 | 15 |
| F1 | 3000 | 6 | F4 | 5500 | 4 | | | | 30 | 30 |
| G2 | 4300 | 6 | G4 | 5300 | 5 | G6 | 8000 | 4 | 45 | 12 |
| H2 | 5100 | 5 | H3 | 6100 | 4 | | | | 45 | 23 |
| I1 | 3000 | 10 | I4 | 5700 | 6 | | | | 45 | 45 |
| J1 | 4400 | 10 | J3 | 8200 | 6 | | | | 75 | 19 |
| K2 | 6000 | 8 | K4 | 6200 | 7 | | | | 75 | 38 |
| L1 | 4400 | 10 | L3 | 5000 | 9 | L5 | 6000 | 8 | 75 | 75 |
| M2 | 5200 | 10 | M4 | 8000 | 7 | | | | 100 | 25 |
| N2 | 5700 | 10 | N6 | 8500 | 8 | | | | 100 | 50 |

to the nearest integer. When the solution time of the BC reaches 3600 s, it means the computing time has stopped at its limit.

The last group of columns in Table 5.7 shows the results from the ALNS. From the 30 runs of each test, we have the best solution objective value found ($z^{\text{BEST}}$), the standard deviation of the solutions' objective values (SD), the total number of runs that ended with no feasible solution (# NF), and also the average solution time in seconds (AT(s)). The column $\text{G}_{\text{ALNS}}$ presents the percentage gap of the best ALNS solution objective value in relation to its LB obtained by the BC. This gap calculation is defined as the one of BC with $z$ substituted with $z^{\text{BEST}}$. Like before, the objective values presented and used to calculate the gaps were rounded to the nearest integer. Moreover, the last line of the table presents some averages.

Table 5.7. Computational results from BC and ALNS

| | | | BC | | | | ALNS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $\Gamma$ | $\beta$ | LB | $z$ | T(s) | $\text{G}_{\text{BC}}$ | $z^{\text{BEST}}$ | SD | # NF | AT(s) | $\text{G}_{\text{ALNS}}$ |
| A2 | 0 | 0 | 164,457 | 164,457 | 0.5 | 0.00% | 164,457 | 0.0% | – | 0.7 | 0.00% |
| | 0 | 750 | 164,457 | 164,457 | 0.8 | 0.00% | 164,457 | 0.4% | – | 0.9 | 0.00% |
| | 2 | 750 | 164,457 | 164,457 | 0.3 | 0.00% | 164,457 | 0.5% | – | 1.0 | 0.00% |
| A4 | 0 | 0 | 136,068 | 136,068 | 0.2 | 0.00% | 136,068 | 1.1% | – | 0.5 | 0.00% |
| | 0 | 750 | 136,581 | 136,581 | 0.2 | 0.00% | 136,581 | 0.1% | – | 0.7 | 0.00% |
| | 2 | 750 | 140,865 | 140,865 | 0.2 | 0.00% | 140,865 | 2.2% | 17 | 0.5 | 0.00% |
| B1 | 0 | 0 | 209,034 | 209,034 | 1.5 | 0.00% | 209,034 | 0.0% | – | 1.0 | 0.00% |
| | 0 | 750 | 209,034 | 209,034 | 1.0 | 0.00% | 209,034 | 0.1% | – | 1.5 | 0.00% |
| | 2 | 750 | 209,034 | 209,034 | 2.3 | 0.00% | 209,089 | 0.2% | – | 1.9 | 0.03% |
| B2 | 0 | 0 | 162,983 | 162,983 | 3.5 | 0.00% | 162,983 | 0.3% | – | 0.7 | 0.00% |
| | 0 | 750 | 162,983 | 162,983 | 1.5 | 0.00% | 162,983 | 0.6% | – | 1.1 | 0.00% |
| | 2 | 750 | 163,431 | 163,431 | 5.9 | 0.00% | 163,473 | 2.0% | 2 | 1.0 | 0.03% |

*continued on next page*

*Table 5.7 continued from previous page*

| | | | BC | | | | ALNS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $\Gamma$ | $\beta$ | LB | $z$ | T(s) | $G_{BC}$ | $z^{BEST}$ | SD | # NF | AT(s) | $G_{ALNS}$ |
| C1 | 0 | 0 | 186,636 | 186,636 | 2.4 | 0.00% | 186,636 | 0.9% | – | 2.2 | 0.00% |
| | 0 | 750 | 187,655 | 187,655 | 6.7 | 0.00% | 187,655 | 1.2% | – | 3.7 | 0.00% |
| | 2 | 750 | 188,842 | 188,842 | 24.3 | 0.00% | 188,842 | 1.9% | – | 4.2 | 0.00% |
| C4 | 0 | 0 | 128,792 | 128,792 | 0.2 | 0.00% | 128,792 | 0.7% | – | 1.6 | 0.00% |
| | 0 | 750 | 131,464 | 131,464 | 0.5 | 0.00% | 131,464 | 1.9% | – | 1.7 | 0.00% |
| | 2 | 750 | 133,580 | 133,580 | 0.2 | 0.00% | 135,682 | 2.6% | 23 | 1.5 | 1.57% |
| D2 | 0 | 0 | 296,077 | 297,853 | 3600.0 | 0.60% | 297,853 | 1.8% | – | 2.3 | 0.60% |
| | 0 | 750 | 297,853 | 297,853 | 2724.5 | 0.00% | 297,853 | 1.5% | – | 4.6 | 0.00% |
| | 2 | 750 | 295,983 | 297,853 | 3600.0 | 0.63% | 297,853 | 1.8% | – | 5.5 | 0.63% |
| D3 | 0 | 0 | 214,017 | 214,017 | 292.8 | 0.00% | 214,017 | 0.2% | – | 1.7 | 0.00% |
| | 0 | 750 | 214,017 | 214,017 | 98.8 | 0.00% | 214,017 | 1.3% | – | 2.5 | 0.00% |
| | 2 | 750 | 214,598 | 214,598 | 294.0 | 0.00% | 214,598 | 1.5% | – | 2.8 | 0.00% |
| E1 | 0 | 0 | 198,239 | 198,239 | 45.5 | 0.00% | 199,005 | 0.9% | – | 1.4 | 0.39% |
| | 0 | 750 | 198,239 | 198,239 | 75.0 | 0.00% | 203,036 | 2.2% | – | 2.3 | 2.42% |
| | 2 | 750 | 198,649 | 198,649 | 41.7 | 0.00% | 217,390 | 2.0% | – | 3.4 | 9.43% |
| E3 | 0 | 0 | 149,576 | 149,576 | 2.3 | 0.00% | 149,576 | 3.1% | – | 2.3 | 0.00% |
| | 0 | 750 | 151,519 | 151,519 | 0.8 | 0.00% | 151,519 | 2.5% | – | 5.8 | 0.00% |
| | 2 | 750 | 154,399 | 154,399 | 2.8 | 0.00% | 154,399 | 3.7% | – | 2.8 | 0.00% |
| F1 | 0 | 0 | 183,723 | 183,723 | 18.5 | 0.00% | 183,723 | 1.7% | – | 4.2 | 0.00% |
| | 0 | 750 | 183,723 | 183,723 | 20.5 | 0.00% | 183,723 | 3.4% | – | 4.4 | 0.00% |
| | 2 | 750 | 183,723 | 183,723 | 23.8 | 0.00% | 183,723 | 3.8% | – | 4.6 | 0.00% |
| F4 | 0 | 0 | 145,456 | 145,456 | 0.1 | 0.00% | 145,456 | 0.9% | – | 2.3 | 0.00% |
| | 0 | 750 | 145,456 | 145,456 | 0.1 | 0.00% | 145,638 | 1.9% | – | 5.8 | 0.13% |
| | 2 | 750 | 145,456 | 145,456 | 0.1 | 0.00% | 147,027 | 1.8% | 16 | 3.9 | 1.08% |
| G2 | 0 | 0 | 225,110 | 225,110 | 80.8 | 0.00% | 225,110 | 1.0% | – | 5.7 | 0.00% |
| | 0 | 750 | 225,110 | 225,110 | 114.8 | 0.00% | 225,663 | 0.9% | – | 5.6 | 0.25% |
| | 2 | 750 | 225,444 | 225,444 | 459.2 | 0.00% | 226,568 | 1.6% | – | 6.2 | 0.50% |
| G4 | 0 | 0 | 202,551 | 202,551 | 1.6 | 0.00% | 202,551 | 0.1% | – | 3.3 | 0.00% |
| | 0 | 750 | 202,551 | 202,551 | 4.3 | 0.00% | 202,551 | 1.4% | – | 3.7 | 0.00% |
| | 2 | 750 | 203,320 | 203,320 | 22.8 | 0.00% | 204,237 | 2.0% | 2 | 4.1 | 0.45% |
| G6 | 0 | 0 | 180,176 | 180,176 | 64.6 | 0.00% | 181,322 | 1.0% | – | 2.6 | 0.64% |
| | 0 | 750 | 180,630 | 180,630 | 60.5 | 0.00% | 180,947 | 2.4% | – | 3.9 | 0.18% |
| | 2 | 750 | 182,081 | 182,081 | 71.4 | 0.00% | 191,603 | 2.7% | 13 | 3.7 | 5.23% |
| H2 | 0 | 0 | 192,446 | 192,446 | 149.9 | 0.00% | 192,904 | 1.1% | – | 4.8 | 0.24% |
| | 0 | 750 | 193,092 | 193,092 | 103.2 | 0.00% | 195,225 | 1.8% | – | 6.9 | 1.10% |
| | 2 | 750 | 197,064 | 197,545 | 3600.1 | 0.24% | 202,268 | 2.0% | – | 10.7 | 2.64% |
| H3 | 0 | 0 | 182,499 | 182,499 | 11.5 | 0.00% | 182,499 | 2.1% | – | 9.6 | 0.00% |
| | 0 | 750 | 186,316 | 186,316 | 26.7 | 0.00% | 187,053 | 1.9% | – | 10.9 | 0.40% |
| | 2 | 750 | 188,727 | 188,727 | 74.9 | 0.00% | 190,855 | 1.7% | – | 16.6 | 1.13% |
| I1 | 0 | 0 | 244,435 | 261,403 | 3600.1 | 6.94% | 256,536 | 1.2% | – | 10.4 | 4.95% |
| | 0 | 750 | 244,074 | 260,212 | 3600.0 | 6.61% | 256,474 | 0.7% | – | 12.1 | 5.08% |
| | 2 | 750 | 244,741 | 262,372 | 3600.0 | 7.20% | 256,717 | 0.9% | – | 24.5 | 4.89% |
| I4 | 0 | 0 | 191,397 | 191,397 | 49.2 | 0.00% | 191,397 | 1.3% | – | 7.8 | 0.00% |

*continued on next page*

*Table 5.7 continued from previous page*

| | | | BC | | | | ALNS | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $\Gamma$ | $\beta$ | LB | $z$ | T(s) | $G_{BC}$ | $z^{BEST}$ | SD | # NF | AT(s) | $G_{ALNS}$ |
| | 0 | 750 | 191,397 | 191,397 | 60.0 | 0.00% | 191,462 | 1.8% | – | 7.4 | 0.03% |
| | 2 | 750 | 191,397 | 191,397 | 46.3 | 0.00% | 191,619 | 2.2% | 1 | 8.8 | 0.12% |
| J1 | 0 | 0 | 274,601 | – | 3600.3 | – | 304,320 | 1.2% | – | 22.6 | 10.82% |
| | 0 | 750 | 273,684 | – | 3600.1 | – | 303,092 | 1.4% | – | 28.1 | 10.75% |
| | 2 | 750 | 275,849 | – | 3600.1 | – | 309,652 | 1.3% | – | 35.5 | 12.25% |
| J3 | 0 | 0 | 216,813 | 245,947 | 3600.2 | 13.44% | 225,100 | 3.9% | – | 14.7 | 3.82% |
| | 0 | 750 | 218,177 | 232,069 | 3600.3 | 6.37% | 226,555 | 3.7% | – | 17.5 | 3.84% |
| | 2 | 750 | 215,679 | 238,870 | 3600.1 | 10.75% | 228,175 | 2.3% | – | 16.1 | 5.79% |
| K2 | 0 | 0 | 264,088 | 327,849 | 3600.4 | 24.14% | 275,513 | 0.9% | – | 24.7 | 4.33% |
| | 0 | 750 | 267,466 | 281,093 | 3600.1 | 5.09% | 275,745 | 1.1% | – | 37.7 | 3.10% |
| | 2 | 750 | 264,448 | 314,323 | 3600.1 | 18.86% | 276,723 | 1.5% | – | 40.3 | 4.64% |
| K4 | 0 | 0 | 245,828 | 269,840 | 3600.1 | 9.77% | 257,710 | 0.8% | – | 17.8 | 4.83% |
| | 0 | 750 | 246,304 | 266,482 | 3600.2 | 8.19% | 258,558 | 1.4% | – | 24.6 | 4.98% |
| | 2 | 750 | 247,350 | 269,572 | 3600.1 | 8.98% | 260,791 | 1.1% | – | 29.6 | 5.43% |
| L1 | 0 | 0 | 266,306 | – | 3600.2 | – | 291,604 | 1.1% | – | 37.6 | 9.50% |
| | 0 | 750 | 268,661 | – | 3600.1 | – | 291,850 | 0.9% | – | 42.4 | 8.63% |
| | 2 | 750 | 268,488 | – | 3600.2 | – | 291,505 | 0.8% | – | 53.4 | 8.57% |
| L3 | 0 | 0 | 254,446 | – | 3600.8 | – | 274,465 | 1.1% | – | 39.6 | 7.87% |
| | 0 | 750 | 253,382 | – | 3600.4 | – | 273,860 | 0.8% | – | 39.0 | 8.08% |
| | 2 | 750 | 251,903 | – | 3600.3 | – | 274,561 | 0.9% | – | 96.3 | 8.99% |
| L5 | 0 | 0 | 236,821 | 287,121 | 3600.3 | 21.24% | 249,581 | 1.7% | – | 34.4 | 5.39% |
| | 0 | 750 | 235,730 | 268,902 | 3600.2 | 14.07% | 249,666 | 2.4% | – | 40.9 | 5.91% |
| | 2 | 750 | 243,111 | 249,420 | 3600.1 | 2.60% | 250,822 | 2.3% | 2 | 37.5 | 3.17% |
| M2 | 0 | 0 | 311,663 | – | 3600.1 | – | 342,540 | 1.0% | – | 52.3 | 9.91% |
| | 0 | 750 | 305,740 | – | 3600.1 | – | 341,859 | 1.8% | – | 61.9 | 11.81% |
| | 2 | 750 | 306,403 | – | 3600.1 | – | 342,164 | 2.6% | – | 69.9 | 11.67% |
| M4 | 0 | 0 | 267,153 | 316,326 | 3600.1 | 18.41% | 282,514 | 1.3% | – | 30.8 | 5.75% |
| | 0 | 750 | 268,303 | 312,923 | 3600.2 | 16.63% | 282,018 | 1.0% | – | 68.7 | 5.11% |
| | 2 | 750 | 266,505 | – | 3600.5 | – | 284,497 | 1.7% | – | 81.5 | 6.75% |
| N2 | 0 | 0 | 300,459 | – | 3600.3 | – | 328,669 | 1.3% | – | 57.5 | 9.39% |
| | 0 | 750 | 298,437 | – | 3600.1 | – | 329,922 | 1.4% | – | 63.3 | 10.55% |
| | 2 | 750 | 302,526 | – | 3600.4 | – | 328,484 | 1.4% | – | 78.8 | 8.58% |
| N6 | 0 | 0 | 251,446 | 332,761 | 3601.0 | 32.34% | 275,535 | 1.1% | – | 29.1 | 9.58% |
| | 0 | 750 | 252,833 | – | 3600.1 | – | 275,533 | 1.2% | – | 36.9 | 8.98% |
| | 2 | 750 | 247,635 | – | 3600.1 | – | 276,690 | 1.9% | – | 45.6 | 11.73% |
| | | | | | 1616.7 | 3.24% | | 1.5% | | 18.5 | 3.27% |

The BC is able to provide optimal solutions for almost all tests in the classes of instances A to H (up to 45 linehaul customers), as denoted by the values of $G_{BC}$. On the other hand, the exact method is unable to find a feasible solution in 18 out of the 90 tests. An interesting observation from the BC results is related to the tests on the class of instances I. These have the same number of linehaul customers as the class of instances H, but almost the double number of backhauls. The difference between instances I1 and

I4 are the number of vehicles, respectively 10 and 6, and vehicle capacities, respectively 3000 and 5700. Yet, the tests with instance I1 do not provide an optimal solution in one hour, while with instance I4, the optimal solution is found in no more than one minute.

The class of instances J have almost the same total number of customers (94) of class I (90), however, the number of linehauls of the J instances represents 167% of the number of linehauls of the I instances. From this point forward (classes J, K, L, M, and N), no optimal guaranteed solutions are found and 18 of the 33 tests failed in returning a feasible solution. It is also noteworthy that among L5 instances, the better $G_{BC}$ gap (2.60%) is presented by the robust version with $\Gamma = 2$ and $\beta = 750$.

Observe that the solutions obtained with ALNS can be considered reasonably stable, as indicated by the the percentage SDs being upper-bounded by 3.9% and with average of 1.5%. On the other hand, there were runs of eight instances that ended without finding a feasible solution, but only for the worst cases, i.e., when $\Gamma = 2$.

The BC found proven optimal solutions in 51 tests, while the ALNS, by its turn, reached solutions with the same objective value in 31 of these tests. On the other hand, calculated when both methods found a feasible solution (which might be optimal), the average gaps of 3.24% for the BC (the $G_{BC}$ average in Table 5.7) and 1.66% for the ALNS (not explicit in the table) denotes that the BC does not favor a general small gap as the ALNS. Among the tests of each instance, the highest $G_{ALNS}$ values are obtained for the worst cases, i.e., the ones with the most revenue deviation allowed (budget of uncertainty $\Gamma = 2$). Furthermore, considering the tests which the BC found a feasible solution without proving its optimality, we have that: in two of these cases occur $G_{BC} < G_{ALNS}$, in other two occur $G_{BC} = G_{ALNS}$, and in the remaining 17 hold that $G_{BC} > G_{ALNS}$.

With respect to the solution time, the ALNS can obtain solutions very fast, where the average of AT(s) is 18.5 s. In fact, even for the larger instances (L, M and N), a solution is found in less than two minutes, on average. On the other hand, the BC average of T(s) is 1616.7 s, being heavily influenced by the 39 tests that ended by time limit. Notwithstanding, the BC finished in less than two minutes in 46 of the tests.

The results on Table 5.7 are evidences that the robust modeling has no more significant intractability than the deterministic modeling. Based on these results, it can be concluded that the exact approach to robust optimization used in this work is efficient to solve the VRPSB under revenue uncertainty.

## 5.5 Conclusions and Future Work

This work presents the first study of a VRPB under uncertainty. More precisely, we consider the problem where the visits to backhauls are optional, each visit has an associated revenue, which is uncertain, and a minimum revenue is required by the solutions. Following a RO approach, the revenues are considered to be bounded by a polyhedron set of uncertainty and a parameter, called budget of uncertainty, is applied to control the size of the uncertainty set. Several different methods to estimate probabilistic bounds for the minimum revenue requirement constraint violation are defined, tested, and compared.

To solve both the deterministic and robust versions of the problem, a BC method was devised and its performance was compared with a heuristic approach.

Several insights could be retrieved from this study. First, solutions are very sensitive to uncertainty, since the structure of the solution changes quickly even for low values of budget of uncertainty. Second, and as expected, the cost of robust solutions is higher than the cost of nominal solutions. In terms of probabilistic bounds, the method that estimates *a posteriori* the probability of constraint violation of a solution is the most rigorous one (since it produces the exact value for that probability), however, it requires the knowledge of the solution structure. A method based on the Irwin-Hall distribution provided tighter bounds than the other *a priori* probabilistic methods, which allows to better characterize the bounds of robust solutions based only on values of budget of uncertainty.

The BC developed in this work finds solutions in a reasonable time for small and medium instances, while the ALNS obtained optimal or good solutions for all instances in a short time. It is also evidenced that the robust approach used in this work created a tractable robust model, where solutions may be obtained with similar computing effort as with the deterministic model.

Different directions for future work on this subject arise. The solution times and gaps obtained suggest an improved BC that uses a cutoff based on the solution provided by the ALNS, what would narrow the solution space to be explored by the exact approach. Besides, considering that few exact methods have been successfully applied to VRPBs, it may be interesting to evaluate more promising solution methods, such as branch-and-price (BP) and BCP algorithms. A consequence of better exact methods is the obtaining of better lower bounds, which would improve the precision of the solution evaluation of heuristic approaches, as done for the ALNS.

Another future direction for research concerns the modeling aspects of the uncertainty. For instance, the uncertainty set may be built using a known distribution or some statistics obtained from historical information, such as the distributionally RO [76]. As such, it could yield tighter uncertainty sets, which may produce better probabilistic bounds for the robust problem and decrease the conservatism of robust solutions. It is worth noting that the complete version of this paper [159] also proposed a chance-constrained model for the VRPSB under uncertainty, including evaluations for a few selected instances. Finally, it would be interesting to apply the robust VRPSB in practice and gather insights from the perspective of integrated logistics.

## Acknowledgments

This work is financed by the ERDF – European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation – COMPETE 2020 Programme and by National Funds through the Portuguese funding agency, FCT – Fundação para a Ciência e a Tecnologia within the project POCI-01-0145-FEDER-016733 (Easyflow).

This research is also financed by Project "TEC4Growth – Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020"

is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

Finally, this research has also been partially supported by CNPq procs. 425340/2016-3 and 314366/2018-0, FAPESP procs. 2015/11937-9, 2016/01860-1 and 2018/08879-5, Fundação Araucária (FA-PR), Secretaria de Estado da Ciência, Tecnologia e Ensino Superior (SETI-PR), Governo do Estado do Paraná (PR), and CAPES.

# Chapter 6

# Final Concluding Remarks

This thesis investigated approaches to three variants of the CVRP that minimize the amount of load-dependent energy consumption and/or impose energy limit per route, namely: CmVRP, CmVRP-LE, and CVRP-LE. We highlight that the energy consumption of vehicle routing is now an established environmental concern [108, 19, 9]. This thesis also explored the resolution of the VRPSB with uncertain revenues, where the rules of backhauls visitation and the uncertainty in the revenues model, for instance, aspects of a real-world application in the wood-base supply chain [184, 5, 6]. In the remaining of this chapter, we recover our main contributions on these problems and point out some possibilities of future research.

## 6.1 Contributions

Several LP relaxations of MILP formulations for the CmVRP were proposed and studied. We proposed the AIL formulation, that is combination of the AI and the AL from the literature, and stated that AIL dominates the formulations it is based on. We also proposed the set partitioning AIL$q$R formulation, that have a pricing subproblem solvable in pseudo-polynomial time, and proved that AIL and AIL$q$R are equivalent. Our experiments show a significant gain in the LP relaxation gap of AIL, when compared to AI and AL. The gap advantage of the AIL may be attributed to the fact that each base formulation alone captures different and complementary aspects of the CmVRP, and putting them together tends to produce a stronger formulation that is worth exploring. The downside is the large amount of time spent in solving the tighter formulation. We also contributed by proposing a somewhat natural way to combine AI with a set partitioning formulation, namely the AI$q$R formulation, which dominates AL. However, such a natural formulation is not likely to be useful in practice since we concluded that its pricing subproblem is strongly NP-hard.

We devised BCP approaches for the CmVRP, CmVRP-LE, and CVRP-LE problems. In these approaches, we modeled the pricing subproblems by RCSPPs with digraphs based on arc-loads, where the usage of the arc-load concept is pivotal in tracking and limiting the energy consumption throughout each route. It was also employed a preprocessing procedure to only keep the arc-loads corresponding to packageable demands, which led

to, in average, 4.16% less arc-loads than a previous approach in the literature. For the CmVRP, we modeled the EIMP by the AL-R formulation, where the best results are yielded with the resource configuration that solely uses the disposable load resource. Comparing this approach to the BCP of Fukasawa et al. [74], in the average, we were able to improve the optimality gap from 0.55% to 0% in a time at least 7 times lower (over 78 instances). Therefore, we improved upon the state-of-the-art exact results for this problem.

Focusing on the problem variants CmVRP-LE and CVRP-LE, it is worth emphasizing that they were proposed in this thesis and that we did not find, in the GVRP literature, a problem that generalizes the CVRP-LE. The EIMP formulations used for the CmVRP-LE is similar to the one of the version without energy limit, while the resource configuration of choice contains, in addition to the disposable load, the non-disposable load and the energy resources. The presence of the non-disposable load resource means that the best performance was achieved by the approach that forbids the creation of labels in which the load resource would be disposed to the detriment of discarding these labels by the dominance rule. Due to the fact that the CVRP-LE's objective function does not consider energy consumption, we were able to use an EIMP formulation given by VF-R, that only has variables per arcs (and not per arc-loads). The best resource configuration for this problem consists of the resources: disposable load, non-disposable load, and energy. In this case, the presence of a non-disposable load avoids an important performance issue intrinsic to this problem: the load excess effect.

The best average times to process the CmVRP, CmVRP-LE, and CVRP-LE are, respectively, 428.2 s, 1029.5 s, and 1335.7 s over a set of 100 instances; which probably indicates a reverse order of hardness among these problems. It is also worth noting that the instances solved in each of these problems has up to 101 vertices, and, for the majority of the instances tested, the variants with limited energy resulted in optimal solutions different from their counterparts without energy limit. We recall that, amid the VRPs that consider load-dependent energy consumption, the enforcing of a limit on the energy consumption of the vehicles by the pricing subproblem constitutes a novelty in the BCP/BP literature.

Considering the approximation algorithms approach, we recall that we slightly improved the analysis of an approximation algorithm for the version of the CmVRP that does not fix the number of vehicles/routes in advance, going from a factor 4 to a factor $4 - \frac{4}{3s^3Q^2}$, where $Q$ is the capacity of the vehicle and $s$ is a scaling factor.

Finally, we also investigated an approach to solve, using RO, the VRPSB with uncertain revenues. In this modeling, the revenues are considered to be bounded by a polyhedron set of uncertainty and a robust parameter – called budget of uncertainty – is applied to control the size of the uncertainty set. To solve both the deterministic and robust versions of the problem, we devised a BC algorithm, whose performance was compared with an heuristic approach. It was possible to observe that, as expected, the cost of robust solutions may be higher than the cost of nominal solutions. In terms of probabilistic bounds of constraint violation, the method that estimates the probability *a posteriori* is the most rigorous one, but it requires a computed solution. On the other hand, a method based on the Irwin-Hall distribution provided tighter probabilistic bounds than

the other *a priori* methods, noting that this method has the advantage of only depend on some instance data and the budget of uncertainty. Our BC algorithm finds solutions in a reasonable time for small and medium instances, while the ALNS obtained optimal or good solutions for all instances in a short time. It is also evidenced that the RO approach created a tractable robust model, where solutions may be obtained with similar computing effort as with the deterministic model.

## 6.2 Future Work

The results on the LP relaxations for the CmVRP point to multiple possibilities of research. The strength of the AIL can be further studied in order to discover its relation with the several families of cuts for the CVRP available in the literature. The experimental results concerning the gaps suggests one may benefit from using AIL in a BC method and AIL$q$R in a BCP method, but further research is needed to speed up their high computational cost. It is also worth searching for a formulation whose LP relaxation *(i)* can be solved in pseudo-polynomial time, and *(ii)* gives bounds that are better than the ones from AIL$q$R.

It might be worth investigating a more fundamental reason that led the BCP approach for the CmVRP-LE to yield the best results with the resource configuration given by the disposable load, non-disposable load, and energy, as this resource option was not expected to be the best for this problem. The instances that we were able to solve for the CmVRP, CmVRP-LE, and CVRP-LE contain at most 1,371,094 arc-loads. Our results suggest that the BCP approaches used should be improved to be applied to instances that imply in higher numbers of arc-loads. One option of research in this line is to improve the removing of arc-loads, by relying, for instance, in the energy limit and branching decisions. Another research possibility is to investigate the usage of cuts specifically devised for arc-load-base formulations, as the ones of Pessoa et al. [141].

Some extended versions of the problems/models with energy considerations, i.e., the CmVRP, CmVRP-LE, and CVRP-LE, might be suitable to be solved by BCP approaches similar to the ones we used. These extensions may consider: the effect of speed variation on the energy consumption, the usage of time windows, and the usage of recharging or refueling stations. In particular, it is worth noting that, even with the energy consumption being generally nonlinear on the speed [19, 75], arc-load approaches offer the potential to consider speed in a linear fashion for these problems, as the optimal speed per arc and load can be pre-computed and incorporated in the costs of arc-loads. On the other hand, the inclusion of time windows in this scenario leads to a more complex situation, as the speed values impact on the time taken to traverse each arc, and thus in the ability to reach vertices within theirs windows [19]. Practical technical specifications could be incorporated into the models, e.g., from vehicles and roads. Bi-objective variants of the problems are also worth investigating, in which one objective may be minimizing distance or time and the other might consists of minimizing energy consumption or greenhouse gas emissions. In addition, exploring heterogeneous fleet and a variable number of vehicles might be of importance, for instance, in environmental-concerned applications. Besides,

it may be worth employing BCP approaches and energy considerations to the VRP with Drones, where each truck carries several drones which, by turn, are dispatched from the truck to perform deliveries [148, 126]. Considering the high potential of practical applications, it is worth exploring variants of the problems with energy considerations that take uncertain parameters into account. The uncertainty may reside in several parameters of the model, among which we highlight two related to energy consumption: *(i)* the energy capacity of the vehicles, and *(ii)* the energy consumption in each arc, which already depends on distance and accumulated load. We point out that the latter item may be suitable to model urban traffic uncertainties. This variants with uncertainty may be treated with RO or other suitable approaches, depending on the stipulated goals.

In the VRPSB with uncertain revenues, one possibility is to use the results with the ALNS algorithm to provide cutoffs to the BC approach. As BCP algorithms have been successfully applied to several VRPs, it is worth approaching this problem with BCP methods with the usage of the VRPSolver framework [143, 158]. Observe that we may maintain the uncertain revenues and the RO treatment with the VRPSolver. In such a research, one may find interesting ways to model the rules of backhaul visiting with the usage of elementarity and packing sets. Another future direction of research concerns the modeling aspects of the uncertainty: for instance, the uncertainty set may be built using a known distribution or some statistics obtained from historical information, such as the distributionally RO [76], which could produce tighter uncertainty sets. Finally, it would be interesting to apply the robust VRPSB in practice and gather insights from the perspective of integrated logistics.

# Bibliography

[1] H. Abeledo, R. Fukasawa, A. Pessoa, and E. Uchoa. The time dependent traveling salesman problem: Polyhedra and algorithm. *Mathematical Programming Computation*, 5(1):27–55, 2013. doi:10.1007/s12532-012-0047-y.

[2] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, and N. Papakostantinou. The complexity of the travelling repairman problem. *RAIRO - Theoretical Informatics and Applications*, 20(1):79–87, 1986. doi:10.1051/ita/1986200100791.

[3] M. Allahviranloo, J. Y. J. Chow, and W. W. Recker. Selective vehicle routing problems under uncertainty without recourse. *Transportation Research Part E: Logistics and Transportation Review*, 62:68–88, 2014. doi:10.1016/j.tre.2013.12.004.

[4] K. Altinkemer and B. Gavish. Heuristics for unequal weight delivery problems with a fixed error guarantee. *Operations Research Letters*, 6(4):149–158, 1987. doi:10.1016/0167-6377(87)90012-5.

[5] P. P. Alvarez and J. R. Vera. Application of robust optimization to the sawmill planning problem. *Annals of Operations Research*, 219(1):457–475, 2014. doi:10.1007/s10479-011-1002-4.

[6] G. Andersson, P. Flisberg, M. Nordström, M. Rönnqvist, and L. Wilhelmsson. A model approach to include wood properties in log sorting and transportation planning. *Information Systems and Operational Research, INFOR*, 54(3):282–303, 2016. doi:10.1080/03155986.2016.1198070.

[7] A. Archer, A. Levin, and D. P. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM Journal on Computing*, 37(5):1472–1498, 2008. doi:10.1137/07068151X.

[8] M. Arenales, V. A. Armentano, R. Morabito, and H. H. Yanasse. *Pesquisa Operacional*. Elsevier, Rio de Janeiro, RJ, BR, 2007.

[9] M. Asghari and S. M. J. Mirzapour Al-e-hashem. New advances in vehicle routing problems: A literature review to explore the future. In H. Derbel, B. Jarboui, and P. Siarry, editors, *Green Transportation and New Advances in Vehicle Routing Problems*, chapter 1, pages 1–42. Springer, Cham, CH, 2020. doi:10.1007/978-3-030-45312-1_1.

[10] M. Asghari and S. M. J. Mirzapour Al-e-hashem. Green vehicle routing problem: A state-of-the-art review. *International Journal of Production Economics*, 231:107899, 2021. doi:10.1016/j.ijpe.2020.107899.

[11] I. Averbakh. On the complexity of a class of combinatorial optimization problems with uncertainty. *Mathematical Programming*, 90(2):263–272, 2001. doi:10.1007/PL00011424.

[12] R. Baldacci, E. Bartolini, and G. Laporte. Some applications of the generalized vehicle routing problem. *Journal of the Operational Research Society*, 61(7):1072–1077, 2010. doi:10.1057/jors.2009.51.

[13] R. Baldacci, N. Christofides, and A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008. doi:10.1007/s10107-007-0178-5.

[14] R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011. doi:10.1287/opre.1110.0975.

[15] R. Baldacci, A. Mingozzi, and R. Roberti. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, 218(1):1–6, 2012. doi:10.1016/j.ejor.2011.07.037.

[16] M. L. Balinski and R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300–304, 1964. doi:10.1287/opre.12.2.300.

[17] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998. doi:10.1287/opre.46.3.316.

[18] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, Inc, Hoboken, NJ, US, 4th edition, 2010.

[19] T. Bektaş and G. Laporte. The pollution-routing problem. *Transportation Research Part B: Methodological*, 45(8):1232–1250, 2011. doi:10.1016/j.trb.2011.02.004.

[20] A. Ben-Tal, L. El Ghaoui, and A. Nemirovskii. *Robust optimization*. Number 0619977. 2009. doi:10.1007/s10957-013-0421-6.

[21] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, US, 1997.

[22] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003. doi:10.1007/s10107-003-0396-4.

[23] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004. doi:10.1287/opre.1030.0065.

[24] D. Bertsimas and A. Thiele. Robust and data-driven optimization: Modern decision making under uncertainty. In *Models, Methods, and Applications for Innovative Decision Making*, INFORMS TutORials in Operations Research, chapter 4, pages 95–122. INFORMS, 2006. doi:10.1287/educ.1063.0022.

[25] D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011. doi:10.1137/080734510.

[26] L. Bianco, A. Mingozzi, and S. Ricciardelli. The traveling salesman problem with cumulative costs. *Networks*, 23(2):81–91, 1993. doi:10.1002/net.3230230202.

[27] J. R. Birge and F. Louveaux. Basic properties and theory. In *Introduction to Stochastic Programming*, Springer Series in Operations Research and Financial Engineering, ORFE, chapter 3, pages 103–161. Springer, New York, NY, US, 2nd edition, 2011. doi:10.1007/978-1-4614-0237-4_3.

[28] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, and M. Sudan. The minimum latency problem. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing, STOC*, pages 163–171, New York, NY, US, 1994. ACM. doi:10.1145/195058.195125.

[29] N. Boysen, S. Fedtke, and S. Schwerdfeger. Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1):1–58, 2021. doi:10.1007/s00291-020-00607-8.

[30] U. Breunig, R. Baldacci, R. F. Hartl, and T. Vidal. The electric two-echelon vehicle routing problem. *Computers & Operations Research*, 103:198–210, 2019. doi:10.1016/j.cor.2018.11.005.

[31] B. P. Bruck, A. G. dos Santos, and J. E. C. Arroyo. Hybrid metaheuristic for the single vehicle routing problem with deliveries and selective pickups. In *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2012. doi:10.1109/cec.2012.6256456.

[32] T. Bulhões, R. Sadykov, and E. Uchoa. A branch-and-price algorithm for the minimum latency problem. *Computers & Operations Research*, 93:66–78, 2018. doi:10.1016/j.cor.2018.01.016.

[33] M. Caramia and P. Dell'Olmo. Multi-objective optimization. In *Multi-objective Management in Freight Logistics: Increasing Capacity, Service Level, Sustainability, and Safety with Optimization Algorithms*, chapter 2, pages 21–51. Springer, Cham, CH, 2020. doi:10.1007/978-3-030-50812-8_2.

[34] D. Carlsson and M. Rönnqvist. Backhauling in forest transportation: models, methods, and practical usage. *Canadian Journal of Forest Research*, 37(12):2612–2623, 2007. doi:10.1139/X07-106.

[35] M. H. Carvalho, M. R. Cerioli, R. Dahab, P. Feofiloff, C. G. Fernandes, C. E. Ferreira, K. S. Guimarães, F. K. Miyazawa, J. C. Pina Jr., J. Soares, and Y. Wabayashi. *Uma Introdução Sucinta a Algoritmos de Aproximação*. Instituto de Matemática Pura e Aplicada, Rio de Janeiro, RJ, BR, 2001.

[36] A. Ceselli, Á. Felipe, M. T. Ortuño, G. Righini, and G. Tirado. A branch-and-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies. *Operations Research Forum*, 2(1):8, 2021. doi:10.1007/s43069-020-00052-x.

[37] M. Chardy and O. Klopfenstein. Handling uncertainties in vehicle routing problems through data preprocessing. *Transportation Research Part E: Logistics and Transportation Review*, 48(3):667–683, 2012. doi:10.1016/j.tre.2011.12.001.

[38] N. Christofides, A. Mingozzi, and P. Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282, 1981. doi:10.1007/BF01589353.

[39] V. Chvátal. *Linear programming*. W H Freeman and Company, New York, NY, US, 1980.

[40] D. Cinar, K. Gakis, and P. M. Pardalos. A 2-phase constructive algorithm for cumulative vehicle routing problems with limited duration. *Expert Systems with Applications*, 56:48–58, 2016. doi:10.1016/j.eswa.2016.02.046.

[41] D. Cinar, B. Cayir Ervural, K. Gakis, and P. M. Pardalos. Constructive algorithms for the cumulative vehicle routing problem with limited duration. In D. Cinar, K. Gakis, and P. M. Pardalos, editors, *Sustainable Logistics and Transportation: Optimization Models and Algorithms*, volume 129 of *SOIA*, pages 57–86. Springer, Cham, CH, 2017. doi:10.1007/978-3-319-69215-9_4.

[42] G. F. Cintra, F. K. Miyazawa, Y. Wakabayashi, and E. C. Xavier. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. *European Journal of Operational Research*, 191(1): 61–85, 2008. doi:10.1016/j.ejor.2007.08.007.

[43] I. M. Coelho, P. L. A. Munhoz, M. N. Haddad, M. J. F. Souza, and L. S. Ochi. A hybrid heuristic based on general variable neighborhood search for the single vehicle routing problem with deliveries and selective pickups. *Electronic Notes in Discrete Mathematics*, 39:99–106, 2012. doi:10.1016/j.endm.2012.10.014.

[44] M. Conforti, G. Cornuéjols, and G. Zambelli. *Integer Programming*. Springer, Cham, CH, 2014. doi:10.1007/978-3-319-11008-0.

[45] R. G. Conrad and M. A. Figliozzi. The recharging vehicle routing problem. In *Proceedings of the Industrial Engineering Research Conference*, 2011. URL https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.646.5645.

[46] C. Contardo and R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014. doi:10.1016/j.disopt.2014.03.001.

[47] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT press, Cambridge, MA, US / London, EN, 3rd edition, 2009.

[48] L. Costa, C. Contardo, and G. Desaulniers. Exact branch-price-and-cut algorithms for vehicle routing. *Transportation Science*, 53(4), 2019. doi:10.1287/trsc.2018.0878.

[49] C. M. Damião, J. M. P. Silva, and E. Uchoa. A branch-cut-and-price algorithm for the cumulative capacitated vehicle routing problem. *4OR*, 2021. doi:10.1007/s10288-021-00498-7.

[50] G. B. Dantzig and J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. doi:10.1287/mnsc.6.1.80.

[51] G. B. Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity analysis of production and allocation*, pages 339–347, 1951.

[52] I. Dayarian, T. G. Crainic, M. Gendreau, and W. Rei. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95:95–123, 2016. doi:10.1016/j.tre.2016.09.004.

[53] T. A. de Queiroz, F. K. Miyazawa, Y. Wakabayashi, and E. C. Xavier. Algorithms for 3d guillotine cutting problems: Unbounded knapsack, cutting stock and strip packing. *Computers & Operations Research*, 39(2):200–212, 2012. doi:10.1016/j.cor.2011.03.011.

[54] E. Demir, T. Bektaş, and G. Laporte. A comparative analysis of several vehicle emission models for road freight transportation. *Transportation Research Part D: Transport and Environment*, 16(5):347–357, 2011. doi:10.1016/j.trd.2011.01.011.

[55] E. Demir, T. Bektaş, and G. Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014. doi:10.1016/j.ejor.2013.12.033.

[56] E. Demir, T. Bektaş, and G. Laporte. The bi-objective pollution-routing problem. *European Journal of Operational Research*, 232(3):464–478, 2014. doi:10.1016/j.ejor.2013.08.002.

[57] U. Derigs, M. Pullmann, U. Vogel, M. Oberscheider, M. Gronalt, and P. Hirsch. Multilevel neighborhood search for solving full truckload routing problems arising in timber transportation. *Electronic Notes in Discrete Mathematics*, 39:281–288, 2012. doi:10.1016/j.endm.2012.10.037.

[58] G. Desaulniers, F. Errico, S. Irnich, and M. Schneider. Exact algorithms for electric vehicle-routing problems with time windows. *Operations Research*, 64(6):1388–1405, 2016. doi:10.1287/opre.2016.1535.

[59] J. Desrosiers and M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, chapter 1, pages 1–32. Springer, Boston, MA, US, 2005. doi:10.1007/0-387-25486-2_1.

[60] J. Desrosiers and M. E. Lübbecke. Branch-price-and-cut algorithms. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Ltd, 2011. doi:10.1002/9780470400531.eorms0118.

[61] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213, 2002. doi:10.1007/s101070100263.

[62] O. Dominguez, D. Guimarans, A. A. Juan, and I. de la Nuez. A biased-randomised large neighbourhood search for the two-dimensional vehicle routing problem with backhauls. *European Journal of Operational Research*, 255(2):442–462, 2016. doi:10.1016/j.ejor.2016.05.002.

[63] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2017. doi:10.1109/TSMC.2016.2582745.

[64] M. Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42(5):977–978, 1994. doi:10.1287/opre.42.5.977.

[65] O. du Merle, D. Villeneuve, J. Desrosiers, and P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1):229–237, 1999. doi:10.1016/S0012-365X(98)00213-1.

[66] R. Eglese and T. Bektaş. Green vehicle routing. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 15, pages 437–458. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch15.

[67] T. Erdelić and T. Carić. A survey on the electric vehicle routing problem: Variants and solution approaches. *Journal of Advanced Transportation*, 2019:5075671, 2019. doi:10.1155/2019/5075671.

[68] S. Erdoğan and E. Miller-Hooks. A green vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 48(1):100–114, 2012. doi:10.1016/j.tre.2011.08.001.

[69] K. Erdoğdu and K. Karabulut. Bi-objective green vehicle routing problem. *International Transactions in Operational Research*, 29(3):1602–1626, 2022. doi:10.1111/itor.13044.

[70] J. Fakcharoenphol, C. Harrelson, and S. Rao. The k-traveling repairman problem. In *Proceedings of the 14$^{th}$ Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 655–664, Philadelphia, PA, US, 2003. Society for Industrial and Applied Mathematics. doi:10.1145/1290672.1290677.

[71] J. Fakcharoenphol, C. Harrelson, and S. Rao. The k-traveling repairmen problem. *ACM Transactions on Algorithms*, 3(4):40:1–40:16, 2007. doi:10.1145/1290672.1290677.

[72] D. Feillet. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR*, 8(4):407–424, 2010. doi:10.1007/s10288-010-0130-z.

[73] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, and R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006. doi:10.1007/s10107-005-0644-x.

[74] R. Fukasawa, Q. He, and Y. Song. A branch-cut-and-price algorithm for the energy minimization vehicle routing problem. *Transportation Science*, 50(1):23–34, 2016. doi:10.1287/trsc.2015.0593.

[75] R. Fukasawa, Q. He, and Y. Song. A disjunctive convex programming approach to the pollution-routing problem. *Transportation Research Part B: Methodological*, 94: 61–79, 2016. doi:https://doi.org/10.1016/j.trb.2016.09.006.

[76] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014. doi:10.1016/j.ejor.2013.09.036.

[77] A. García-Nájera, J. A. Bullinaria, and M. A. Gutiérrez-Andrade. An evolutionary approach for multi-objective vehicle routing problems with backhauls. *Computers & Industrial Engineering*, 81:90–108, 2015. doi:10.1016/j.cie.2014.12.029.

[78] W. W. Garvin, H. W. Crandall, J. B. John, and R. A. Spellman. Applications of linear programming in the oil industry. *Management Science*, 3(4):407–430, 1957. doi:10.1287/mnsc.3.4.407.

[79] D. R. Gaur and R. R. Singh. A heuristic for cumulative vehicle routing using column generation. *Discrete Applied Mathematics*, 228:140–157, 2017. doi:10.1016/j.dam.2016.05.030.

[80] D. R. Gaur, A. Mudgal, and R. R. Singh. Routing vehicles to minimize fuel consumption. *Operations Research Letters*, 41(6):576–580, 2013. doi:10.1016/j.orl.2013.07.007.

[81] D. R. Gaur, A. Mudgal, and R. R. Singh. Approximation algorithms for cumulative VRP with stochastic demands. In *Proceedings of the 2$^{nd}$ International Conference on Algorithms and Discrete Applied Mathematics, CALDAM*, volume 9602 of *LNCS*, pages 176–189, Cham, CH, 2016. Springer. doi:10.1007/978-3-319-29221-2_15.

[82] D. R. Gaur, A. Mudgal, and R. R. Singh. Improved approximation algorithms for cumulative VRP with stochastic demands. *Discrete Applied Mathematics*, 2018. doi:10.1016/j.dam.2018.01.012.

[83] M. Gendreau and J.-Y. Potvin, editors. *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science, ISOR*. Springer, Boston, MA, US, 2nd edition, 2010. doi:10.1007/978-1-4419-1665-5.

[84] V. Ghilas, E. Demir, and T. van Woensel. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines. *Computers & Operations Research*, 72:12–30, 2016. doi:10.1016/j.cor.2016.01.018.

[85] M. T. Godinho, L. Gouveia, and T. L. Magnanti. Combined route capacity and route length models for unit demand vehicle routing problems. *Discrete Optimization*, 5 (2):350–372, 2008. doi:10.1016/j.disopt.2007.05.001.

[86] D. Goeke and M. Schneider. Routing a mixed fleet of electric and conventional vehicles. *European Journal of Operational Research*, 245(1):81–99, 2015. doi:10.1016/j.ejor.2015.01.049.

[87] M. Goetschalckx and C. Jacobs-Blecha. The vehicle routing problem with backhauls. *European Journal of Operational Research*, 42(1):39–51, 1989. doi:10.1016/0377-2217(89)90057-X.

[88] B. Golden, S. Raghavan, and E. Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *ORCS*. Springer, Boston, MA, US, 2008. doi:10.1007/978-0-387-77778-8.

[89] B. L. Gorissen, I. Yanikoğlu, and D. den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015. doi:10.1016/j.omega.2014.12.006.

[90] M. Granada-Echeverri, E. M. Toro, and J. J. Santa. A mixed integer linear programming formulation for the vehicle routing problem with backhauls. *International Journal of Industrial Engineering Computations*, 10:295–308, 2019. doi:10.5267/j.ijiec.2018.6.003.

[91] I. E. Grossmann, R. M. Apap, B. A. Calfa, P. García-Herreros, and Q. Zhang. Recent advances in mathematical programming techniques for the optimization of process systems under uncertainty. *Computers & Chemical Engineering*, 91:3–14, 2016. doi:10.1016/j.compchemeng.2016.03.002.

[92] G. Gutiérrez-Jarpa, V. Marianov, and C. Obreque. A single vehicle routing problem with fixed delivery and optional collections. *IIE Transactions*, 41(12):1067–1079, 2009. doi:10.1080/07408170903113771.

[93] Y. A. Guzman, L. R. Matthews, and C. A. Floudas. New a priori and a posteriori probabilistic bounds for robust counterpart optimization: I. unknown probability distributions. *Computers & Chemical Engineering*, 84:568–598, 2016. doi:10.1016/j.compchemeng.2015.09.014.

[94] Y. A. Guzman, L. R. Matthews, and C. A. Floudas. New a priori and a posteriori probabilistic bounds for robust counterpart optimization: II. a priori bounds for known symmetric and asymmetric probability distributions. *Computers & Chemical Engineering*, 101:279–311, 2017. doi:10.1016/j.compchemeng.2016.07.002.

[95] Y. A. Guzman, L. R. Matthews, and C. A. Floudas. New a priori and a posteriori probabilistic bounds for robust counterpart optimization: III. exact and near-exact a posteriori expressions for known probability distributions. *Computers & Chemical Engineering*, 103:116–143, 2017. doi:10.1016/j.compchemeng.2017.03.001.

[96] M. Haimovich and A. H. G. Rinnooy Kan. Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research*, 10(4):527–542, 1985. URL http://www.jstor.org/stable/3689422.

[97] M. Hassanalian and A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 2017. doi:10.1016/j.paerosci.2017.04.003.

[98] H. Hernández-Pérez and J.-J. Salazar-González. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 196(3):987–995, 2009. doi:10.1016/j.ejor.2008.05.009.

[99] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *European Journal of Operational Research*, 252(3):995–1018, 2016. doi:10.1016/j.ejor.2016.01.038.

[100] P. Hirsch. Minimizing empty truck loads in round timber transport with tabu search strategies. *International Journal of Information Systems and Supply Chain Management*, 4(2):15–41, 2011. doi:10.4018/jisscm.2011040102.

[101] P. Hokama, F. K. Miyazawa, and E. C. Xavier. A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, 47:1–13, 2016. doi:10.1016/j.eswa.2015.10.013.

[102] S. Irnich and G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, and M. M. Solomon, editors, *Column Generation*, chapter 2, pages 33–65. Springer, Boston, MA, US, 2005. doi:10.1007/0-387-25486-2_2.

[103] S. Irnich, P. Toth, and D. Vigo. The family of vehicle routing problems. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 1, pages 1–33. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch1.

[104] R. Jothi and B. Raghavachari. Approximating the k-traveling repairman problem with repairtimes. *Journal of Discrete Algorithms*, 5(2):293–303, 2007. doi:10.1016/j.jda.2006.03.023.

[105] Ö. Kabadurmuş, M. S. Erdoğan, Y. Özkan, and M. Köseoğlu. A multi-objective solution of green vehicle routing problem. *Logistics, Supply Chain, Sustainability and Global Challenges*, 10(1):31–44, 2019. doi:10.2478/jlst-2019-0003.

[106] S.-C. Kang, T. S. Brisimi, and I. C. Paschalidis. Distribution-dependent robust linear optimization with applications to inventory control. *Annals of Operations Research*, 231(1):229–263, 2015. doi:10.1007/s10479-013-1467-4.

[107] İ. Kara, B. Y. Kara, and M. K. Yetiş. Energy minimizing vehicle routing problem. In *Proceedings of the 1st International Conference on Combinatorial Optimization and Applications, COCOA*, volume 4616 of *LNCS*, pages 62–71, Berlin/Heidelberg, DE, 2007. Springer. doi:10.1007/978-3-540-73556-4_9.

[108] İ. Kara, B. Y. Kara, and M. K. Yetiş. Cumulative vehicle routing problems. In T. Caric and H. Gold, editors, *Vehicle Routing Problem*, chapter 6. IntechOpen, Rijeka, HR, 2008. doi:10.5772/5812.

[109] Ç. Koç and G. Laporte. Vehicle routing with backhauls: Review and research perspectives. *Computers & Operations Research*, 91:79–91, 2018. doi:10.1016/j.cor.2017.11.003.

[110] R. Kramer, A. Subramanian, T. Vidal, and L. d. A. F. Cabral. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015. doi:10.1016/j.ejor.2014.12.009.

[111] I. Kucukoglu, R. Dewil, and D. Cattrysse. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161: 107650, 2021. doi:10.1016/j.cie.2021.107650.

[112] R. Kumar, A. Unnikrishnan, and S. T. Waller. Capacitated-vehicle routing problem with backhauls on trees. *Transportation Research Record: Journal of the Transportation Research Board*, 2263(1):92–102, 2011. doi:10.3141/2263-11.

[113] A. Langevin, F. Soumis, and J. Desrosiers. Classification of travelling salesman problem formulations. *Operations Research Letters*, 9(2):127–132, 1990. doi:10.1016/0167-6377(90)90052-7.

[114] G. Laporte and Y. Nobert. A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum*, 5(2):77–85, 1983. doi:10.1007/BF01720015.

[115] A. N. Letchford and J.-J. Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105(2-3):251–274, 2006. doi:10.1007/s10107-005-0652-x.

[116] A. N. Letchford and J.-J. Salazar-González. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730–738, 2015. doi:10.1016/j.ejor.2015.02.028.

[117] A. N. Letchford and J.-J. Salazar-González. The capacitated vehicle routing problem: Stronger bounds in pseudo-polynomial time. *European Journal of Operational Research*, 272(1):24–31, 2019. doi:10.1016/j.ejor.2018.06.002.

[118] B. Li, D. Krushinsky, T. van Woensel, and H. A. Reijers. An adaptive large neighborhood search heuristic for the share-a-ride problem. *Computers & Operations Research*, 66:170–180, 2016. doi:10.1016/j.cor.2015.08.008.

[119] P. Li, H. Arellano-Garcia, and G. Wozny. Chance constrained programming approach to process optimization under uncertainty. *Computers & Chemical Engineering*, 32(1):25–45, 2008. doi:10.1016/j.compchemeng.2007.05.009.

[120] Y. Li, H. Chen, and C. Prins. Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. *European Journal of Operational Research*, 252(1):27–38, 2016. doi:10.1016/j.ejor.2015.12.032.

[121] Z. Li, R. Ding, and C. A. Floudas. A comparative theoretical and computational study on robust counterpart optimization: I. robust linear optimization and robust mixed integer linear optimization. *Industrial & Engineering Chemistry Research*, 50(18):10567–10603, 2011. doi:10.1021/ie200150p.

[122] S.-C. Liu and C.-H. Chung. A heuristic method for the vehicle routing problem with backhauls and inventory. *Journal of Intelligent Manufacturing*, 20(1):29, 2008. doi:10.1007/s10845-008-0101-9.

[123] M. E. Lübbecke. Column generation. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Ltd, 2011. doi:10.1002/9780470400531.eorms0158.

[124] A. Lucena. Time-dependent traveling salesman problem–the deliveryman case. *Networks*, 20(6):753–763, 1990. doi:10.1002/net.3230200605.

[125] J. Lysgaard, A. N. Letchford, and R. W. Eglese. A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2): 423–445, 2004. doi:10.1007/s10107-003-0481-8.

[126] G. Macrina, L. Di Puglia Pugliese, F. Guerriero, and G. Laporte. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, 120:102762, 2020. doi:10.1016/j.trc.2020.102762.

[127] J. E. Marengo, D. L. Farnsworth, and L. Stefanic. A geometric derivation of the irwin-hall distribution. *International Journal of Mathematics and Mathematical Sciences*, 2017:3571419, 2017. doi:10.1155/2017/3571419.

[128] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementations*. John Wiley & Sons, Ltd, New York, NY, US, 1990.

[129] R. Masson, F. Lehuédé, and O. Péton. An adaptive large neighborhood search for the pickup and delivery problem with transfers. *Transportation Science*, 47(3): 344–355, 2013. doi:10.1287/trsc.1120.0432.

[130] I. Méndez-Díaz, P. Zabala, and A. Lucena. A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, 156(17):3223–3237, 2008. doi:10.1016/j.dam.2008.05.009.

[131] F. K. Miyazawa and C. C. de Souza. Introdução à otimização combinatória. In *Anais da 34ª Jornada de Atualização em Informática – XXXV Anais do Congresso da Sociedade Brasileira de Computação*, volume 1, pages 123–190. SBC, Porto Alegre, RS, BR, 2015.

[132] M. H. Mulati and F. K. Miyazawa. Tighter analysis of an approximation for the cumulative VRP. In *Anais do II Encontro de Teoria da Computação – XXXVII Congresso da Sociedade Brasileira de Computação, ETC*, pages 150–153, Porto Alegre, RS, BR, 2017. SBC. doi:10.5753/etc.2017.3202.

[133] M. H. Mulati, R. Fukasawa, and F. K. Miyazawa. The arc-item-load and related formulations for the cumulative vehicle routing problem. *Discrete Optimization*, 45: 100710, 2022. doi:10.1016/j.disopt.2022.100710.

[134] F. Ordóñez. Robust vehicle routing. In J. J. Hasenbein, editor, *Risk and Optimization in an Uncertain World*, INFORMS TutORials in Operations Research, chapter 7, pages 153–178. INFORMS, 2010. doi:10.1287/educ.1100.0078.

[135] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018. doi:10.1002/net.21818.

[136] C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Dover, Mineola, NY, US, 1998.

[137] D. Pecin, A. Pessoa, M. Poggi, and E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017. doi:10.1007/s12532-016-0108-8.

[138] D. Pecin, A. Pessoa, M. Poggi, E. Uchoa, and H. Santos. Limited memory rank-1 cuts for vehicle routing problems. *Operations Research Letters*, 45(3):206–209, 2017. doi:10.1016/j.orl.2017.02.006.

[139] S. Pelletier, O. Jabali, and G. Laporte. 50th anniversary invited article—goods distribution with electric vehicles: Review and research perspectives. *Transportation Science*, 50(1):3–22, 2016. doi:10.1287/trsc.2015.0646.

[140] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018. doi:10.1287/ijoc.2017.0784.

[141] A. Pessoa, M. P. de Aragão, and E. Uchoa. Robust branch-cut-and-price algorithms for vehicle routing problems. In E. Golden, Bruce and Raghavan, S. and Wasil, editor, *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *ORCS*, pages 297–325. Springer, Boston, MA, US, 2008. doi:10.1007/978-0-387-77778-8_14.

[142] A. Pessoa, R. Sadykov, and E. Uchoa. Enhanced branch-cut-and-price algorithm for heterogeneous fleet vehicle routing problems. *European Journal of Operational Research*, 270(2):530–543, 2018. doi:10.1016/j.ejor.2018.04.009.

[143] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 2020. doi:10.1007/s10107-020-01523-z.

[144] J.-C. Picard and M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110, 1978. doi:10.1287/opre.26.1.86.

[145] D. Pisinger and S. Ropke. Large neighborhood search. In M. Gendreau and J.-Y. Potvin, editors, *Handbook of Metaheuristics*, volume 146 of *International Series in Operations Research & Management Science, ISOR*, chapter 13, pages 399–419. Springer, Boston, MA, US, 2nd edition, 2010. doi:10.1007/978-1-4419-1665-5_13.

[146] M. Poggi and E. Uchoa. New exact algorithms for the capacitated vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 3, pages 59–86. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch3.

[147] M. Poggi de Aragão and E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. In *Proceedings of the Conference Mathematical Program in Rio: A Conference in Honour of Nelson Maculan*, pages 56–61, 2003. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.1278.

[148] S. Poikonen, X. Wang, and B. Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017. doi:10.1002/net.21746.

[149] L. Pradenas, B. Oportus, and V. Parada. Mitigation of greenhouse gas emissions in vehicle routing problems with backhauling. *Expert Systems with Applications*, 40 (8):2985–2991, 2013. doi:10.1016/j.eswa.2012.12.014.

[150] J. Privé, J. Renaud, F. Boctor, and G. Laporte. Solving a vehicle-routing problem arising in soft-drink distribution. *Journal of the Operational Research Society*, 57 (9):1045–1052, 2006. doi:10.1057/palgrave.jors.2602087.

[151] L. D. P. Pugliese and F. Guerriero. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200, 2013. doi:10.1002/net.21511.

[152] E. Queiroga, Y. Frota, R. Sadykov, A. Subramanian, E. Uchoa, and T. Vidal. On the exact solution of vehicle routing problems with backhauls. *European Journal of Operational Research*, 287(1):76–89, 2020. doi:10.1016/j.ejor.2020.04.047.

[153] R. Roberti and A. Mingozzi. Dynamic ng-path relaxation for the delivery man problem. *Transportation Science*, 48(3):413–424, 2014. doi:10.1287/trsc.2013.0474.

[154] S. Ropke and D. Pisinger. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4): 455–472, 2006. doi:10.1287/trsc.1050.0135.

[155] R. Sadykov. Private communication, 2022.

[156] R. Sadykov and F. Vanderbeck. Bapcod - a generic branch-and-price code. Technical report, Inria Bordeaux Sud-Ouest, 2021. URL https://hal.inria.fr/hal-03340548.

[157] R. Sadykov, F. Vanderbeck, A. Pessoa, I. Tahiri, and E. Uchoa. Primal heuristics for branch and price: The assets of diving methods. *INFORMS Journal on Computing*, 31(2):251–267, 2019. doi:10.1287/ijoc.2018.0822.

[158] R. Sadykov, E. Uchoa, and A. Pessoa. A bucket graph–based labeling algorithm with application to vehicle routing. *Transportation Science*, 55(1):4–28, 2021. doi:10.1287/trsc.2020.0985.

[159] M. J. Santos, E. Curcio, M. H. Mulati, P. Amorim, and F. K. Miyazawa. A robust optimization approach for the vehicle routing problem with selective backhauls. *Transportation Research Part E: Logistics and Transportation Review*, 136:101888, 2020. doi:10.1016/j.tre.2020.101888.

[160] D. Schermer, M. Moeini, and O. Wendt. Algorithms for solving the vehicle routing problem with drones. In *Proceedings of the 10$^{th}$ Asian Conference on Intelligent Information and Database Systems, ACIIDS*, volume 10751 of *LNCS*, pages 352–361, Cham, CH, 2018. Springer. doi:10.1007/978-3-319-75417-8_33.

[161] M. Schneider, A. Stenger, and D. Goeke. The electric vehicle-routing problem with time windows and recharging stations. *Transportation Science*, 48(4):500–520, 2014. doi:10.1287/trsc.2013.0490.

[162] F. Semet, P. Toth, and D. Vigo. Classical exact algorithms for the capacitated vehicle routing problem. In P. Toth and D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 2, pages 37–57. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2$^{nd}$ edition, 2014. doi:10.1137/1.9781611973594.ch2.

[163] P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Proceedings of the 4$^{th}$ International Conference on Principles and Practice of Constraint Programming, CP*, volume 1520 of *LNCS*, pages 417–431, Berlin/Heidelberg, DE, 1998. Springer. doi:10.1007/3-540-49481-2_30.

[164] R. R. Singh and D. R. Gaur. Cumulative VRP: A simplified model of green vehicle routing. In D. Cinar, K. Gakis, and P. M. Pardalos, editors, *Sustainable Logistics and Transportation: Optimization Models and Algorithms*, volume 129 of *SOIA*, pages 39–55. Springer, Cham, CH, 2017. doi:10.1007/978-3-319-69215-9_3.

[165] J. C. Smith and S. Ahmed. Introduction to robust optimization. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Ltd, 2011. doi:10.1002/9780470400531.eorms0427.

[166] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973. doi:10.1287/opre.21.5.1154.

[167] A. Subramanian and E. Queiroga. Solution strategies for the vehicle routing problem with backhauls. *Optimization Letters*, 14(8):2429–2441, 2020. doi:10.1007/s11590-020-01564-5.

[168] H. Süral and J. H. Bookbinder. The single-vehicle routing problem with unrestricted backhauls. *Networks*, 41(3):127–136, 2003. doi:10.1002/net.10067.

[169] P. Toth and D. Vigo. An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science*, 31(4):372–385, 1997. doi:10.1287/trsc.31.4.372.

[170] P. Toth and D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.

[171] G. Y. Tütüncü, C. Carreto, and B. Baker. A visual interactive approach to classical and mixed vehicle routing problems with backhauls. *Omega*, 37(1):138–154, 2009. doi:10.1016/j.omega.2006.11.001.

[172] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, and A. Subramanian. New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858, 2017. doi:10.1016/j.ejor.2016.08.012.

[173] V. V. Vazirani. *Approximation Algorithms*. Springer, Berlin/Heidelberg, DE, 2002.

[174] S. Wang, M. Liu, and F. Chu. Approximate and exact algorithms for an energy minimization traveling salesman problem. *Journal of Cleaner Production*, 249:119433, 2020. doi:10.1016/j.jclepro.2019.119433.

[175] X. Wang, S. Poikonen, and B. Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697, 2017. doi:10.1007/s11590-016-1035-3.

[176] Z. Wang and J.-B. Sheu. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, 122:350–364, 2019. doi:10.1016/j.trb.2019.03.005.

[177] D. P. Williamson and D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, New York, NY, US, 2011.

[178] L. A. Wolsey. *Integer Programming*. John Wiley & Sons, Ltd, 2nd edition, 2020. doi:10.1002/9781119606475.

[179] Z. Wu and J. Zhang. A branch-and-price algorithm for two-echelon electric vehicle routing problem. *Complex & Intelligent Systems*, 2021. doi:10.1007/s40747-021-00403-z.

[180] Y. Xiao, Q. Zhao, I. Kaku, and Y. Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7):1419–1431, 2012. doi:10.1016/j.cor.2011.08.013.

[181] C. A. Yano, T. J. Chan, L. K. Richter, T. Cutler, K. G. Murty, and D. McGettigan. Vehicle routing at quality stores. *Interfaces*, 17(2):52–63, 1987. doi:10.1287/inte.17.2.52.

[182] E. E. Zachariadis, C. D. Tarantilis, and C. T. Kiranoudis. The load-dependent vehicle routing problem and its pick-up and delivery extension. *Transportation Research Part B: Methodological*, 71:158–181, 2015. doi:10.1016/j.trb.2014.11.004.

[183] S. Zajac. An adaptive large neighborhood search for the periodic vehicle routing problem. In *Proceedings of the 8th International Conference on Computational Logistics, ICCL*, volume 10572 of *LNCS*, pages 34–48, Cham, CH, 2017. Springer. doi:10.1007/978-3-319-68496-3_3.

[184] M. K. Zanjani, D. Ait-Kadi, and M. Nourelfath. Robust production planning in a manufacturing environment with random yield: A case in sawmill production planning. *European Journal of Operational Research*, 201(3):882–891, 2010. doi:10.1016/j.ejor.2009.03.041.

[185] S. Zhang, Y. Gajpal, S. S. Appadoo, and M. M. S. Abdulkader. Electric vehicle routing problem with recharging stations for minimizing energy consumption. *International Journal of Production Economics*, 203:404–413, 2018. doi:10.1016/j.ijpe.2018.07.016.

# Appendix A

# Full Experimental Results of the Relaxed Arc-Item, Arc-Load, and Arc-Item-Load Formulations for the CmVRP

We start by noting that this appendix is related to Chapter 2. In Table A.1 we present the full experimental results that are condensed in Tables 2.4 and 2.5 in Subsection 2.5.4. The definitions of the columns are presented in Subsection 2.5.3. It is worth emphasizing that the original values (reported on Table A.1) of vehicle capacity $Q$ of some instances were preprocessed by being divided by a scaling factor, resulting in $Q'$, as in the following.

- E-n13-k4: $Q' = 6000/100 = 60$;
- E-n22-k4: $Q' = 6000/100 = 60$;
- E-n30-k3: $Q' = 4500/25 = 180$; and
- P-n22-k8: $Q' = 3000/100 = 30$.

Table A.1. AI, AL, and AIL relaxed models full results with regular instances

| | | Arc-Item | | | | Arc-Load | | | | Arc-Item-Load | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ # | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) |
| A-n32-k5 | 100 | 1.1 | 39,989.20 | | 2.7 | 4.4 | 39,696.83 | | 3.4 | 709.9 | 40,706.20 | | 0.9 |
| A-n33-k5 | 100 | 1.0 | 31,501.52 | | 5.6 | 4.1 | 32,539.13 | | 2.5 | 1,459.8 | 32,851.38 | | 1.5 |
| A-n33-k6 | 100 | 1.2 | 34,983.37 | | 5.5 | 3.9 | 35,797.29 | | 3.3 | 3,486.5 | 36,739.79 | | 0.8 |
| A-n34-k5 | 100 | 2.1 | 37,227.31 | | 9.5 | 8.1 | 38,553.10 | | 6.3 | 4,229.3 | 39,967.98 | | 2.8 |
| A-n36-k5 | 100 | 2.9 | 37,098.70 | | 5.6 | 19.5 | 38,517.66 | | 2 | 2,146.2 | 38,804.80 | | 1.2 |
| A-n37-k5 | 100 | 2.1 | 29,641.05 | | 4.4 | 21.6 | 29,820.92 | | 3.8 | 3,032.1 | 30,716.29 | | 0.9 |
| A-n37-k6 | 100 | 4.1 | 45,749.12 | | 8.7 | 11.9 | 48,001.58 | | 4.2 | 1,427.9 | 48,559.70 | | 3.1 |
| A-n38-k5 | 100 | 2.3 | 33,965.74 | | 11.9 | 13.2 | 34,656.54 | | 10.1 | 12,753.8 | 36,363.12 | | 5.7 |
| A-n39-k5 | 100 | 8.5 | 39,528.51 | | 7.5 | 27.4 | 41,663.94 | | 2.5 | 4,124.5 | 42,058.45 | | 1.6 |
| A-n39-k6 | 100 | 1.9 | 38,660.46 | | 6.5 | 5.7 | 40,253.84 | | 2.7 | 2,420.7 | 41,069.33 | | 0.7 |
| A-n44-k6 | 100 | 10.2 | 45,311.43 | | 5.4 | 15.8 | 47,329.91 | | 1.1 | 7,345.8 | 47,755.22 | | 0.3 |
| A-n45-k6 | 100 | 7.5 | 45,499.19 | | 12.4 | 15.7 | 48,706.45 | | 6.3 | 9,757.0 | 49,802.76 | | 4.1 |
| A-n45-k7 | 100 | 15.2 | 58,971.15 | | 5.3 | 11.9 | 60,871.63 | | 2.2 | 11,519.9 | 61,363.06 | | 1.4 |
| A-n46-k7 | 100 | 11.6 | 45,784.79 | | 4.7 | 18.6 | 47,458.33 | | 1.2 | 9,597.5 | 47,927.25 | | 0.2 |
| A-n48-k7 | 100 | 23.6 | 54,841.82 | | 6.2 | 21.8 | 57,077.72 | | 2.4 | 17,725.9 | 57,315.46 | | 2 |
| A-n53-k7 | 100 | 33.6 | 50,902.54 | | 8.8 | 38.0 | 53,035.17 | | 4.9 | 34,624.5 | 54,194.67 | | 2.9 |
| A-n54-k7 | 100 | 56.1 | 59,115.59 | | 7.2 | 31.8 | 62,061.14 | | 2.6 | 65,921.4 | 62,936.50 | | 1.2 |
| A-n55-k9 | 100 | 13.0 | 54,660.22 | | 6.2 | 19.1 | 55,825.30 | | 4.2 | 32,250.2 | 56,877.71 | | 2.4 |
| A-n60-k9 | 100 | 83.1 | 68,282.56 | | 8.1 | 39.4 | 72,488.48 | | 2.5 | 81,763.9 | 72,838.46 | | 2 |

*continued on next page*

*Table A.1 continued from previous page*

| | | Arc-Item | | | | Arc-Load | | | | Arc-Item-Load | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ # | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) |
| A | 19 | 14.8 | | | 7 | 17.5 | | | 3.6 | 16,120.9 | | | 1.9 |
| B-n31-k5 | 100 | 1.3 | 33,827.11 | | 3.9 | 3.4 | 33,331.46 | | 5.3 | 2,519.7 | 34,590.88 | | 1.8 |
| B-n34-k5 | 100 | 1.5 | 40,083.49 | | 9.2 | 6.5 | 41,236.60 | | 6.6 | 7,022.0 | 42,959.26 | | 2.7 |
| B-n35-k5 | 100 | 1.1 | 47,969.38 | | 9.3 | 8.9 | 48,167.65 | | 8.9 | 885.9 | 48,947.42 | | 7.5 |
| B-n38-k6 | 100 | 2.6 | 39,353.87 | | 4.4 | 5.0 | 39,854.33 | | 3.2 | 3,765.7 | 40,216.89 | | 2.3 |
| B-n39-k5 | 100 | 1.6 | 27,112.00 | | 7.4 | 5.4 | 27,085.61 | | 7.5 | 11,870.4 | 28,178.21 | | 3.7 |
| B-n41-k6 | 100 | 2.4 | 41,653.42 | | 10.1 | 7.0 | 44,088.08 | | 4.8 | 4,617.0 | 45,211.06 | | 2.4 |
| B-n43-k6 | 100 | 11.4 | 36,572.45 | | 8.6 | 12.3 | 38,261.48 | | 4.4 | 11,224.6 | 39,022.54 | | 2.5 |
| B-n44-k7 | 100 | 3.7 | 47,587.28 | | 8.4 | 7.9 | 49,995.45 | | 3.8 | 21,155.5 | 50,535.65 | | 2.7 |
| B-n45-k5 | 100 | 5.2 | 33,457.47 | | 10.2 | 12.5 | 34,316.92 | | 7.9 | 23,516.7 | 35,164.71 | | 5.6 |
| B-n45-k6 | 100 | 9.8 | 34,623.87 | | 11.2 | 28.0 | 35,143.17 | | 9.9 | 35,988.5 | 37,661.95 | | 3.4 |
| B-n50-k7 | 100 | 3.1 | 36,603.71 | | 7.7 | 25.8 | 37,440.41 | | 5.6 | 68,479.1 | 38,008.92 | | 4.2 |
| B-n50-k8 | 100 | 58.2 | 65,631.61 | | 8.6 | 35.2 | 70,858.26 | | 1.3 | 67,816.9 | 70,980.11 | | 1.1 |
| B-n51-k7 | 100 | 8.4 | 52,660.93 | | 22.2 | 34.3 | 54,949.21 | | 18.9 | 66,034.3 | 56,097.61 | | 17.2 |
| B-n52-k7 | 100 | 13.2 | 36,592.44 | | 8.7 | 51.9 | 37,140.49 | | 7.3 | 59,713.7 | 38,575.42 | | 3.7 |
| B-n67-k10 | 100 | 121.8 | 53,073.96 | | 6.7 | 36.8 | 54,429.07 | | 4.4 | 77,429.0 | 54,943.50 | | 3.5 |
| B | 15 | 16.4 | | | 9.1 | 18.7 | | | 6.7 | 30,802.6 | | | 4.3 |
| E-n13-k4 | 6000 | 0.0 | 728,517.91 | | 6.5 | 0.0 | 774,300.00 | | 0.7 | 0.4 | 778,066.67 | | 0.2 |
| E-n22-k4 | 6000 | 0.2 | 1,131,170.00 | | 5.4 | 0.2 | 1,178,581.54 | | 1.4 | 59.6 | 1,195,200.00 | * | 0 |
| E-n30-k3 | 4500 | 1.0 | 1,003,393.75 | | 19.6 | 15.5 | 1,071,232.83 | | 14.1 | 9,316.0 | 1,105,568.11 | | 11.4 |
| E-n31-k7 | 140 | 4.1 | 23,255.65 | | 9.1 | 5.7 | 25,327.38 | | 1 | 800.0 | 25,516.77 | | 0.3 |
| E | 4 | 1.3 | | | 10.2 | 5.4 | | | 4.3 | 2,544.0 | | 1 | 3 |
| P-n16-k8 | 35 | 0.0 | 8,389.56 | | 0.5 | 0.0 | 8,434.50 | * | 0 | 0.1 | 8,434.50 | * | 0 |
| P-n19-k2 | 160 | 0.1 | 15,644.75 | | 17.5 | 2.0 | 17,352.67 | | 8.5 | 325.1 | 18,022.95 | | 5 |
| P-n20-k2 | 160 | 0.2 | 16,052.25 | | 17.9 | 3.3 | 18,137.00 | | 7.3 | 604.3 | 18,847.06 | | 3.7 |
| P-n21-k2 | 160 | 0.3 | 15,757.56 | | 17.5 | 3.3 | 17,803.14 | | 6.8 | 673.0 | 18,523.52 | | 3.1 |
| P-n22-k2 | 160 | 0.4 | 16,116.74 | | 17.9 | 4.3 | 18,541.17 | | 5.6 | 1,132.2 | 19,214.07 | | 2.1 |
| P-n22-k8 | 3000 | 0.1 | 914,620.43 | | 7.9 | 0.0 | 978,408.33 | | 1.5 | 2.1 | 985,706.28 | | 0.7 |
| P-n23-k8 | 40 | 0.1 | 11,016.91 | | 8.7 | 0.0 | 11,818.40 | | 2.1 | 3.0 | 11,818.40 | | 2.1 |
| P-n40-k5 | 140 | 8.8 | 28,967.73 | | 11.9 | 18.0 | 31,872.52 | | 3.1 | 12,347.6 | 32,321.42 | | 1.7 |
| P-n45-k5 | 150 | 15.4 | 34,035.61 | | 13.7 | 52.9 | 38,121.14 | | 3.4 | 28,895.4 | 38,874.85 | | 1.5 |
| P-n50-k7 | 150 | 15.6 | 39,058.48 | | 7.3 | 81.4 | 41,640.86 | | 1.1 | 40,342.6 | 42,070.63 | | 0.1 |
| P-n50-k8 | 120 | 12.9 | 36,205.92 | | 10.4 | 70.1 | 38,815.08 | | 4 | 15,501.5 | 38,965.98 | | 3.6 |
| P-n50-k10 | 100 | 4.3 | 34,219.82 | | 5.9 | 16.4 | 35,506.50 | | 2.4 | 7,054.7 | 35,682.50 | | 1.9 |
| P-n51-k10 | 80 | 8.6 | 29,397.99 | | 6.6 | 17.8 | 30,581.66 | | 2.9 | 6,844.1 | 30,663.97 | | 2.6 |
| P-n55-k7 | 170 | 25.1 | 43,962.25 | | 9.6 | 133.0 | 47,344.28 | | 2.6 | 81,615.1 | 48,199.70 | | 0.9 |
| P-n55-k8 | 160 | 15.2 | 43,072.41 | | 7.1 | 143.0 | 45,304.51 | | 2.3 | 75,211.6 | 45,756.31 | | 1.4 |
| P-n55-k10 | 115 | 14.6 | 38,583.67 | | 4.7 | 29.7 | 39,913.55 | | 1.5 | 11,349.7 | 40,164.92 | | 0.8 |
| P-n55-k15 | 70 | 8.1 | 33,966.00 | | 9 | 3.0 | 36,382.37 | | 2.5 | 8,582.2 | 36,512.91 | | 2.1 |
| P-n60-k10 | 120 | 21.8 | 43,579.44 | | 6 | 83.5 | 45,265.45 | | 2.4 | 56,595.7 | 45,814.96 | | 1.2 |
| P-n60-k15 | 80 | 10.6 | 39,354.71 | | 3.9 | 8.8 | 40,412.26 | | 1.3 | 11,544.0 | 40,519.06 | | 1 |
| P-n65-k10 | 130 | 68.9 | 49,740.25 | | 6.1 | 88.4 | 51,819.86 | | 2.2 | 85,160.6 | 52,353.50 | | 1.2 |
| P | 20 | 11.6 | | | 9.5 | 37.9 | | 1 | 3.2 | 22,189.2 | | 1 | 1.8 |
| ulysses-n16-k3 | 5 | 0.0 | 18,566.48 | | 8.1 | 0.0 | 19,482.88 | | 3.6 | 0.6 | 19,994.65 | | 1.1 |
| gr-n17-k3 | 6 | 0.0 | 7,265.66 | | 0.9 | 0.0 | 7,331.20 | * | 0 | 0.4 | 7,331.20 | * | 0 |
| gr-n21-k3 | 7 | 0.1 | 11,756.40 | * | 0 | 0.0 | 11,756.40 | * | 0 | 0.5 | 11,756.40 | * | 0 |
| ulysses-n22-k4 | 6 | 0.2 | 22,919.50 | | 5.8 | 0.0 | 23,202.42 | | 4.6 | 1.6 | 23,682.88 | | 2.6 |
| gr-n24-k4 | 7 | 0.7 | 6,899.19 | | 3.4 | 0.0 | 7,081.20 | | 0.8 | 7.0 | 7,093.50 | | 0.7 |
| fri-n26-k3 | 10 | 0.8 | 6,377.11 | | 7.5 | 0.0 | 6,704.69 | | 2.8 | 15.6 | 6,887.02 | | 0.1 |
| bayg-n29-k4 | 8 | 0.7 | 7,598.30 | | 12.1 | 0.0 | 8,334.38 | | 3.6 | 30.6 | 8,422.20 | | 2.6 |
| bays-n29-k5 | 6 | 0.9 | 8,560.17 | | 8 | 0.0 | 8,975.16 | | 3.5 | 12.1 | 9,011.33 | | 3.2 |
| dantzig-n42-k4 | 11 | 46.1 | 5,780.02 | | 9.2 | 0.1 | 6,165.44 | | 3.1 | 818.2 | 6,279.93 | | 1.3 |
| swiss-n42-k5 | 9 | 7.2 | 6,935.52 | | 11.5 | 0.1 | 7,392.63 | | 5.6 | 526.4 | 7,622.63 | | 2.7 |
| gr-n48-k3 | 16 | 117.7 | 40,772.70 | | 22.9 | 0.6 | 46,124.96 | | 12.8 | 3,899.9 | 48,122.04 | | 9 |
| att-n48-k4 | 15 | 34.3 | 256,206.82 | | 17.1 | 0.3 | 281,740.34 | | 8.8 | 5,043.2 | 294,974.97 | | 4.6 |
| hk-n48-k4 | 15 | 224.5 | 97,256.11 | | 16.4 | 0.4 | 107,752.58 | | 7.4 | 3,435.1 | 110,324.01 | | 5.1 |
| V | 13 | 33.3 | | 1 | 9.5 | 0.1 | | 2 | 4.4 | 1,060.9 | | 2 | 2.5 |
| | 71 | 16.8 | | 1 | 8.8 | 19.6 | | 3 | 4.3 | 17,409.7 | | 4 | 2.6 |

# Appendix B

# Detailed Experimental Results of the Best BCP Approaches for the CmVRP, CmVRP-LE, and CVRP-LE

We start by noting that this appendix is related to Chapter 3. Detailed experimental results of the best BCP approaches for the CmVRP, CmVRP-LE, and CVRP-LE problems, over the base sample set of instances, are present in tables B.1, B.2, and B.3. Besides fields previously presented in Section 3.5, these tables also contains the column "Stat", whose content possibilities and their meaning are given as follows: "OPT" – an optimal solution was found; "TLN" – the time limit was reached with no feasible solution found; and "IF" – the instance model is reported as infeasible.

Table B.1. Detailed results of the BCP algorithm for the CmVRP using the AL-R formulation with the $\tilde{Q}_M^+$ resource over the base sample set of instances

| Inst | $Q$ # | Stat | #O | #S | #N | #X | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n33-k6 | 100 | OPT | | | | | 0 | 0 | 9.2 | 9.2 | 1.0 | 5.1 | 0.4 | 0.0 | 0.0 |
| A-n38-k5 | 100 | OPT | | | | | 0 | 0 | 27.0 | 27.0 | 1.0 | 13.6 | 4.4 | 0.1 | 0.0 |
| A-n45-k7 | 100 | OPT | | | | | 0 | 0 | 23.7 | 23.7 | 1.0 | 13.0 | 1.3 | 0.1 | 0.0 |
| A-n55-k9 | 100 | OPT | | | | | 0 | 0 | 46.3 | 46.2 | 1.0 | 24.9 | 3.5 | 0.1 | 0.0 |
| A-n63-k10 | 100 | OPT | | | | | 0 | 0.04 | 346.3 | 248.7 | 3.0 | 261.3 | 31.8 | 3.8 | 64.2 |
| A | 5 | | 5 | 0 | 0 | 0 | 0 | 0.01 | 90.5 | 71.0 | 1.4 | 63.6 | 8.3 | 0.8 | 12.8 |
| B-n35-k5 | 100 | OPT | | | | | 0 | 0 | 284.4 | 284.4 | 1.0 | 69.5 | 199.5 | 5.7 | 0.0 |
| B-n44-k7 | 100 | OPT | | | | | 0 | 0 | 32.0 | 31.9 | 1.0 | 14.6 | 9.2 | 0.1 | 0.0 |
| B-n51-k7 | 100 | OPT | | | | | 0 | 1.74 | 6,701.9 | 359.0 | 29.0 | 2,515.8 | 3,590.9 | 251.3 | 640.6 |
| B-n63-k10 | 100 | OPT | | | | | 0 | 0 | 161.2 | 161.2 | 1.0 | 66.6 | 49.2 | 0.3 | 0.0 |
| B | 4 | | 4 | 0 | 0 | 0 | 0 | 0.43 | 1,794.9 | 209.1 | 8.0 | 666.6 | 962.2 | 64.4 | 160.2 |
| E-n23-k3 | 4500 | OPT | | | | | 0 | 0 | 401.4 | 401.4 | 1.0 | 323.8 | 0.0 | 0.0 | 0.0 |
| E-n31-k7 | 140 | OPT | | | | | 0 | 0 | 8.0 | 8.0 | 1.0 | 3.7 | 0.0 | 0.0 | 0.0 |
| E-n76-k8 | 180 | OPT | | | | | 0 | 0 | 376.6 | 376.6 | 1.0 | 244.6 | 53.0 | 0.8 | 0.0 |
| E | 3 | | 3 | 0 | 0 | 0 | 0 | 0 | 262.0 | 262.0 | 1.0 | 190.7 | 17.7 | 0.3 | 0.0 |
| P-n20-k2 | 160 | OPT | | | | | 0 | 0 | 8.0 | 8.0 | 1.0 | 4.4 | 0.6 | 0.0 | 0.0 |
| P-n40-k5 | 140 | OPT | | | | | 0 | 0 | 23.2 | 23.1 | 1.0 | 12.8 | 1.5 | 0.1 | 0.0 |
| P-n51-k10 | 80 | OPT | | | | | 0 | 0 | 19.5 | 19.5 | 1.0 | 9.6 | 1.2 | 0.1 | 0.0 |
| P-n60-k10 | 120 | OPT | | | | | 0 | 0 | 36.7 | 36.7 | 1.0 | 19.5 | 1.9 | 0.1 | 0.0 |
| P | 4 | | 4 | 0 | 0 | 0 | 0 | 0 | 21.8 | 21.8 | 1.0 | 11.6 | 1.3 | 0.1 | 0.0 |
| ulysses-n16-k3 | 5 | OPT | | | | | 0 | 0 | 1.3 | 1.3 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| gr-n24-k4 | 7 | OPT | | | | | 0 | 0 | 1.5 | 1.5 | 1.0 | 0.1 | 0.0 | 0.0 | 0.0 |
| swiss-n42-k5 | 9 | OPT | | | | | 0 | 0 | 2.7 | 2.7 | 1.0 | 0.9 | 0.1 | 0.1 | 0.0 |
| V | 3 | | 3 | 0 | 0 | 0 | 0 | 0 | 1.9 | 1.9 | 1.0 | 0.3 | 0.1 | 0.0 | 0.0 |
| | 19 | | 19 | 0 | 0 | 0 | 0 | 0.09 | 447.9 | 109.0 | 2.6 | 189.7 | 207.8 | 13.8 | 37.1 |

Table B.2. Detailed results of the BCP algorithm for the CmVRP-LE using the AL-R formulation with the $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ resource over the base sample set of instances

| Inst | $Q$ # | Stat | #O | #IF | #S | #N | #X | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n33-k6 | 100 | OPT | | | | | | 0 | 0 | 30.0 | 30.0 | 1.0 | 15.7 | 8.8 | 0.1 | 0.0 |
| A-n38-k5 | 100 | OPT | | | | | | 0 | 0 | 105.9 | 105.9 | 1.0 | 80.6 | 15.8 | 0.8 | 0.0 |
| A-n45-k7 | 100 | OPT | | | | | | 0 | 0.37 | 285.3 | 231.6 | 3.0 | 187.1 | 76.5 | 3.5 | 41.2 |
| A-n55-k9 | 100 | OPT | | | | | | 0 | 0 | 250.7 | 250.7 | 1.0 | 223.0 | 9.4 | 1.0 | 0.0 |
| A-n63-k10 | 100 | OPT | | | | | | 0 | 0.59 | 870.7 | 547.1 | 7.0 | 646.1 | 58.5 | 79.3 | 200.2 |
| A | 5 | | 5 | 0 | 0 | 0 | 0 | 0 | 0.19 | 308.5 | 233.1 | 2.6 | 230.5 | 33.8 | 16.9 | 48.3 |
| B-n35-k5 | 100 | OPT | | | | | | 0 | 0 | 458.3 | 458.3 | 1.0 | 260.2 | 179.6 | 10.6 | 0.0 |
| B-n44-k7 | 100 | OPT | | | | | | 0 | 0.56 | 368.1 | 139.1 | 5.0 | 291.6 | 35.5 | 14.2 | 96.2 |
| B-n51-k7 | 100 | TLN | | | | | | | | 7,225.3 | 577.3 | 19.0 | 3,386.5 | 3,462.9 | 191.6 | 744.4 |
| B-n63-k10 | 100 | OPT | | | | | | 0 | 0.44 | 1,022.6 | 682.8 | 7.0 | 787.3 | 82.0 | 30.8 | 152.5 |
| B | 4 | | 3 | 0 | 0 | 1 | 0 | 0 | 0.33 | 2,268.6 | 464.4 | 8.0 | 1,181.4 | 940.0 | 61.8 | 248.3 |
| E-n23-k3 | 4500 | OPT | | | | | | 0 | 0 | 883.7 | 883.7 | 1.0 | 720.9 | 87.6 | 0.1 | 0.0 |
| E-n31-k7 | 140 | IF | | | | | | | | 22.5 | 22.5 | 1.0 | 18.2 | 0.0 | 0.0 | 0.0 |
| E-n76-k8 | 180 | OPT | | | | | | 0 | 0 | 1,741.4 | 1741.4 | 1.0 | 1,506.2 | 159.0 | 1.0 | 0.0 |
| E | 3 | | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 882.5 | 882.5 | 1.0 | 748.4 | 82.2 | 0.4 | 0.0 |
| P-n20-k2 | 160 | OPT | | | | | | 0 | 0 | 24.1 | 24.1 | 1.0 | 11.1 | 9.9 | 0.0 | 0.0 |
| P-n40-k5 | 140 | OPT | | | | | | 0 | 0 | 117.5 | 117.5 | 1.0 | 91.8 | 15.2 | 0.2 | 0.0 |
| P-n51-k10 | 80 | OPT | | | | | | 0 | 0 | 65.4 | 65.4 | 1.0 | 38.4 | 13.2 | 0.2 | 0.0 |
| P-n60-k10 | 120 | OPT | | | | | | 0 | 0 | 191.8 | 191.7 | 1.0 | 156.2 | 9.3 | 0.4 | 0.0 |
| P | 4 | | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 99.7 | 99.7 | 1.0 | 74.4 | 11.9 | 0.2 | 0.0 |
| ulysses-n16-k3 | 5 | OPT | | | | | | 0 | 0 | 1.2 | 1.2 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| gr-n24-k4 | 7 | OPT | | | | | | 0 | 0 | 2.1 | 2.1 | 1.0 | 0.4 | 0.4 | 0.0 | 0.0 |
| swiss-n42-k5 | 9 | OPT | | | | | | 0 | 0 | 7.0 | 7.0 | 1.0 | 4.6 | 0.5 | 0.1 | 0.0 |
| V | 3 | | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 3.5 | 3.5 | 1.0 | 1.7 | 0.3 | 0.1 | 0.0 |
| | 19 | | 17 | 1 | 0 | 1 | 0 | 0 | 0.12 | 719.7 | 320.0 | 2.9 | 443.5 | 222.3 | 17.6 | 65.0 |

Table B.3. Detailed results of BCP algorithm for the CVRP-LE using the VF-R formulation with the $\tilde{Q}_M^+ + \tilde{Q}_s^- + \tilde{E}_s$ resource over the base sample set of instances

| Inst | $Q$ # | Stat | #O | #IF | #S | #N | #X | G(%) | R(%) | T(s) | R T(s) | # Nds | CG T(s) | RCF & CE T(s) | LP T(s) | SB T(s) |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n33-k6 | 100 | OPT | | | | | | 0 | 0 | 19.2 | 19.2 | 1.0 | 11.5 | 2.2 | 0.1 | 0.0 |
| A-n38-k5 | 100 | OPT | | | | | | 0 | 0 | 76.7 | 76.7 | 1.0 | 55.8 | 16.0 | 0.4 | 0.0 |
| A-n45-k7 | 100 | OPT | | | | | | 0 | 0.31 | 154.8 | 116.7 | 5.0 | 80.0 | 31.3 | 11.4 | 15.8 |
| A-n55-k9 | 100 | OPT | | | | | | 0 | 0 | 114.7 | 114.7 | 1.0 | 92.5 | 12.5 | 0.6 | 0.0 |
| A-n63-k10 | 100 | OPT | | | | | | 0 | 0 | 808.1 | 775.8 | 3.0 | 654.8 | 120.1 | 12.1 | 32.1 |
| A | 5 | | 5 | 0 | 0 | 0 | 0 | 0 | 0.06 | 234.7 | 220.6 | 2.2 | 178.9 | 36.4 | 4.9 | 9.6 |
| B-n35-k5 | 100 | OPT | | | | | | 0 | 0.93 | 731.2 | 621.0 | 5.0 | 254.1 | 344.9 | 66.4 | 57.4 |
| B-n44-k7 | 100 | OPT | | | | | | 0 | 0 | 71.3 | 71.3 | 1.0 | 60.6 | 5.3 | 0.1 | 0.0 |
| B-n51-k7 | 100 | TLN | | | | | | | | 7,214.1 | 656.1 | 33.0 | 5,511.8 | 0.0 | 1572.0 | 1539.5 |
| B-n63-k10 | 100 | OPT | | | | | | 0 | 0.82 | 880.0 | 564.4 | 17.0 | 506.0 | 81.7 | 132.0 | 153.6 |
| B | 4 | | 3 | 0 | 0 | 1 | 0 | 0 | 0.59 | 2,224.2 | 478.2 | 14.0 | 1,583.1 | 108.0 | 442.6 | 437.6 |
| E-n23-k3 | 4500 | OPT | | | | | | 0 | 0 | 1,617.9 | 1617.9 | 1.0 | 1,425.0 | 126.1 | 0.0 | 0.0 |
| E-n31-k7 | 140 | IF | | | | | | | | 17.3 | 17.3 | 1.0 | 14.1 | 0.0 | 0.0 | 0.0 |
| E-n76-k8 | 180 | OPT | | | | | | 0 | 0 | 2,514.6 | 2514.6 | 1.0 | 1,998.4 | 447.1 | 3.6 | 0.0 |
| E | 3 | | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1,383.3 | 1383.3 | 1.0 | 1,145.8 | 191.1 | 1.2 | 0.0 |
| P-n20-k2 | 160 | OPT | | | | | | 0 | 0 | 28.9 | 28.9 | 1.0 | 18.8 | 7.8 | 0.0 | 0.0 |
| P-n40-k5 | 140 | OPT | | | | | | 0 | 0 | 75.8 | 75.8 | 1.0 | 68.9 | 0.0 | 0.1 | 0.0 |
| P-n51-k10 | 80 | OPT | | | | | | 0 | 0 | 40.7 | 40.7 | 1.0 | 25.2 | 7.1 | 0.2 | 0.0 |
| P-n60-k10 | 120 | OPT | | | | | | 0 | 0 | 146.8 | 146.5 | 1.0 | 102.9 | 26.7 | 0.3 | 0.0 |
| P | 4 | | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 73.1 | 73.0 | 1.0 | 53.9 | 10.4 | 0.2 | 0.0 |
| ulysses-n16-k3 | 5 | OPT | | | | | | 0 | 0 | 1.1 | 1.1 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| gr-n24-k4 | 7 | OPT | | | | | | 0 | 0 | 1.9 | 1.9 | 1.0 | 0.3 | 0.2 | 0.0 | 0.0 |
| swiss-n42-k5 | 9 | OPT | | | | | | 0 | 0.33 | 20.7 | 17.3 | 3.0 | 6.5 | 6.7 | 1.6 | 1.9 |
| V | 3 | | 3 | 0 | 0 | 0 | 0 | 0 | 0.11 | 7.9 | 6.8 | 1.7 | 2.3 | 2.3 | 0.5 | 0.6 |
| | 19 | | 17 | 1 | 0 | 1 | 0 | 0 | 0.14 | 765.0 | 393.6 | 4.2 | 573.0 | 65.0 | 94.8 | 94.8 |