UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA DE PRODUÇÃO

FERNANDO RIBEIRO DE SENNA

A STUDY OF TWO-ECHELON ROUTING PROBLEMS APPLIED TO LAST-MILE DELIVERY: FORMULATIONS AND EXACT METHODS

SÃO CARLOS -SP
2024

FERNANDO RIBEIRO DE SENNA

A STUDY OF TWO-ECHELON ROUTING PROBLEMS APPLIED TO LAST-MILE DELIVERY: FORMUTIONS AND EXACT METHODS

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de São Carlos, para obtenção do título de mestre em Engenharia de Produção.

Orientador: Prof. Dr. Reinaldo Morabito
Coorientador: Prof. Dr. Pedro Munari

São Carlos-SP
2024

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Defesa de Dissertação de Mestrado do candidato Fernando Ribeiro de Senna, realizada em 09/08/2024.

Comissão Julgadora:

Prof. Dr. Reinaldo Morabito Neto (UFSCar)

Prof. Dr. Pedro Augusto Munari Junior (UFSCar)

Profa. Dra. Kelly Cristina Poldi (UNICAMP)

Prof. Dr. Leandro Callegari Coelho (ULaval)

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia de Produção.

À minha família.

Acknowledgements

For the completion of this master's degree, I had the support of many people. In particular, I would like to thank:

Prof. Reinaldo Morabito, my advisor, not only for providing the necessary guidance and support for my research but also for giving me numerous insights into my professional life, patiently discussing repeatedly the same topics in our eternal "coffee breaks".

Prof. Pedro Munari, my co-advisor, who, on top of being essential for the technical aspects of my research, was always available to discuss a wide range of topics, besides being a great company at various conferences and events.

Prof. Leandro C. Coelho for warmly welcoming me in Quebec and integrating me into his research laboratory as if I were one of his regular students, providing all the material support I needed and closely overseeing my research, almost as a second co-advisor for this master's.

Prof. Antônio Carlos Moretti for being a key figure at the beginning of my academic and professional career, introducing me to Applied Mathematics and Operations Research. If it were not for him, this dissertation would certainly be defended in another university and possibly in a completely different area, such as biomathematics or theoretical physics.

Prof. Douglas Alem for hosting me in Edinburgh and providing an excellent research internship, expanding my horizons and allowing me to delve deeper into Operations Research applications in logistics from a different perspective than the one I was accustomed to.

Prof. Kelly Poldi for carefully reading my dissertation despite the limited deadline and giving interesting suggestions that made this text much better.

Profs. Anand Subramanian, Helio Fuchigami, Jean-François Côté, Maristela Santos, Maryam Darvish, and Mateus Martin who were always welcoming and supportive during various academic events and interactions.

Lucas Duarte and Robson Santos from the Graduate Program in Production Engineering's office for their helpful and efficient assistance in resolving any administrative demands I had.

Pierre Marchand for helping me during my stay in Quebec with all necessary administrative support and for being very welcoming, even including me in the CIRRELT beach volleyball.

Sandra, Carlos, Thaís, and Otávio for warmly receiving me in São Carlos and making me feel part of the family.

My friends from the laboratories and classes, Alberto Locatelli, Alex Abreu, Arineia Nogueira, Bahareh Naderizand, Carlo Soverchia, Davide Porta, Eduardo Becker, Eduardo

Sanches, Eliass Fennich, Guilherme Chagas, Imadeddine Aziez, Ishaan Maheshwari, Kamyla Ferreira, Maria Vitória Bussacarini, Nadja Oliveira (and Mateus), Quentin Marissiaux, Rafael Ajudarte, Razieh Mousavi, Rodrigo Ramalho, Tarley Mansur, and Yure Rocha, for their assistance at various times and, most importantly, for making my master's experiences much more enjoyable, both in São Carlos and Quebec.

My parents, Marcus and Tânia, for always ensuring I had access to the best possible education, in the broadest sense, not just academically.

My sisters, Laís and Ana Laura, and my brother-in-law, Victor, for always being the best companions.

My grandmother, Deborah, for always being present in my life regardless of the distance, and my aunt, Suzi, for always encouraging my academic adventures.

My relatives and friends for making life much more enjoyable.

Finally, I would like to thank FAPESP and CAPES for their financial support towards the completion of my master's degree. This dissertation was supported by the São Paulo Research Foundation (FAPESP), grants nº 2021/14441-5 and 2022/09679-5, and by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 1.

Agradecimentos

Para a realização deste mestrado contei com o apoio de muitas pessoas. Em especial, gostaria de agradecer:

Ao meu orientador, Prof. Reinaldo Morabito, não só por ter dado as diretrizes e o apoio necessários à minha pesquisa, mas também por ter me dado diversas orientações para a minha vida profissional, com muita paciência para discutir o mesmo assunto diversas vezes nos nossos eternos "cafezinhos".

Ao meu coorientador, Prof. Pedro Munari, que, além de ter sido essencial para a parte técnica da minha pesquisa, estava sempre à disposição para discutir os temas mais variados possíveis, além de ter sido ótima companhia em diversos congressos e eventos.

Ao Prof. Leandro C. Coelho, por ter me recebido em Quebec de forma tão calorosa e me incluído em seu laboratório de pesquisa como se fosse um de seus alunos regulares, me fornecendo todo o apoio material de que precisava, além de ter acompanhado minha pesquisa tão de perto que poderia ser considerado um segundo coorientador desse mestrado.

Ao Prof. Antônio Carlos Moretti, por ter sido figura essencial no início de minha carreira acadêmica e profissional, me apresentando a Matemática Aplicada e a Pesquisa Operacional. Não fosse por ele, essa dissertação certamente seria defendida em outra universidade e, talvez, em uma área completamente diferente, como biomatemática ou física teórica.

Ao Prof. Douglas Alem, por ter me recebido em Edimburgo e proporcionado um excelente estágio em pesquisa, expandindo meus horizontes e possibilitando que eu me aprofundasse na área de Pesquisa Operacional aplicada à logística com uma perspectiva diferente da que eu estava acostumado.

À Profa. Kelly Poldi por ter lido minha dissertação atentamente apesar do curto prazo e por ter dado sugestões interessantes que melhoraram muito este texto.

Aos Profs. Anand Subramanian, Helio Fuchigami, Jean-François Côté, Maristela Santos, Maryam Darvish e Mateus Martin, que, nos diversos eventos acadêmicos de que participei e nas demais interações, sempre foram extremamente receptivos e acolhedores.

Ao Lucas Duarte e ao Robson Santos da secretaria do Programa de Pós-Graduação em Engenharia de Produção, por serem sempre muito prestativos e eficientes em resolver todas e quaisquer demandas administrativas que eu tivesse.

Ao Pierre Marchand, por ter me ajudado em minha estadia em Quebec com todo o apoio administrativo de que precisasse e por ter sido extremamente acolhedor, me incluindo até nos

jogos de vôlei do CIRRELT.

À Sandra, ao Carlos, à Thaís e ao Otávio, por terem me recebido em São Carlos tão carinhosamente e me acolhido como parte da família.

Aos amigos de laboratório e disciplinas, Alberto Locatelli, Alex Abreu, Arineia Nogueira, Bahareh Naderizand, Carlo Soverchia, Davide Porta, Eduardo Becker, Eduardo Sanches, Eliass Fennich, Guilherme Chagas, Imadeddine Aziez, Ishaan Maheshwari, Kamyla Ferreira, Maria Vitória Bussacarini, Nadja Oliveira (e Mateus), Quentin Marissiaux, Rafael Ajudarte, Razieh Mousavi, Rodrigo Ramalho, Tarley Mansur e Yure Rocha, por terem me ajudado em diversos momentos e, mais importante, por terem tornado minhas experiências no mestrado muito mais divertidas, seja em São Carlos ou em Quebec.

Aos meus pais, Marcus e Tânia, por sempre terem se preocupado em garantir que eu tivesse acesso à melhor educação possível, no sentido mais amplo da palavra, não só academicamente.

Às minhas irmãs, Laís e Ana Laura, e ao meu cunhado, Victor, por serem sempre as melhores companhias.

À minha avó, Deborah, por estar sempre presente em todos os momentos, não importando a distância. À minha tia, Suzi, por sempre ter me incentivado nas minhas aventuras acadêmicas.

Aos meus familiares e amigos, por tornarem a vida muito mais divertida.

Por fim, agradeço à FAPESP e à CAPES pelo apoio financeiro à realização do meu mestrado. Essa dissertação foi financiada pela Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), processos nº 2021/14441-5 e 2022/09679-5, e pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 1.

Abstract

Vehicle routing in urban environments encompasses additional challenges for logistics services providers due to traffic conditions, city regulations, and difficulty in finding parking locations. To overcome these issues, companies usually adopt alternative delivery systems, such as the ones that are reflected in two-echelon routing problems. These problems are based on the idea of having larger vehicles taking goods from depots to intermediary facilities or parking locations, and smaller vehicles or walking deliverymen taking goods from these points to the final customers. Two examples of such problems are the two-echelon location-routing problem (2E-LRP) and the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD). The first encompasses decisions on vehicle routing in both echelons and the facilities to be opened or used. The latter considers that vehicles may carry more than one deliveryman to increase vehicle efficiency. In this dissertation, we propose improvements for both of these problems, focusing on formulations, valid inequalities, Benders decomposition schemes, and exact solution methods. For the 2E-LRP, two novel formulations are presented and compared to the benchmark one. Their linear programming relaxations and their performance under a mixed-integer programming solver are compared, showing that the novel formulations greatly outperform the benchmark of the literature. Also, 125 new best known lower bounds and 55 new optimal solutions were found for the 131 benchmark instances evaluated. Regarding the VRPTWMD, two realistic extensions of the problem were proposed, incorporating in the optimization the deliveryman routes and the decision on which customers to serve in each vehicle stop, which are usually considered to be preprocessed in the literature. Valid inequalities and Benders decomposition schemes were proposed to develop exact solution algorithms for these new variants of the VRPTWMD. Managerial insights show the importance of applying these variants instead of the common approach from the literature, leading to cost reductions of around 10%.

Keywords: vehicle routing; last-mile delivery; two-echelon location-routing problem; multiple deliverymen; Benders decomposition.

Resumo

O roteamento de veículos em ambientes urbanos engloba desafios adicionais para provedores de serviços logísticos devido às condições de tráfego, regulamentações das cidades e dificuldade de encontrar locais para estacionar. Para superar essas dificuldades, empresas comumente adotam sistemas de entrega alternativos, como os que são refletidos em problemas de roteamento em dois níveis. Esses problemas se baseiam na ideia de ter veículos maiores transportando bens de depósitos para facilidades intermediárias ou pontos de estacionamento, e veículos menores ou entregadores a pé transportando bens desses pontos até os consumidores finais. Dois exemplos de tais problemas são o problema de localização-roteamento em dois níveis (PLR-2N) e o problema de roteamento de veículos com janelas de tempo e múltiplos entregadores (PRVJTME). O primeiro envolve decisões sobre roteamento de veículos em dois níveis e quais facilidades devem ser abertas. O último considera que veículos podem carregar mais de um entregador para aumentar a eficiência dos veículos. Nesta dissertação, propõem-se melhorias para ambos os problemas, focando-se em formulações, desigualdades válidas, decomposições de Benders e métodos exatos de solução. Para o PLR-2N, duas novas formulações são apresentadas e comparadas com o padrão da literatura. Suas relaxações lineares e seus desempenhos quando aplicadas a um *solver* de programação inteira mista são comparados, mostrando que as novas formulações têm desempenho muito melhor. Ainda, 125 novos melhores limitantes inferiores e 55 novas soluções ótimas foram encontrados dentre as 131 instâncias da literatura avaliadas. Considerando-se o PRVJTME, duas extensões realistas do problema foram propostas, incorporando-se na otimização as rotas dos entregadores e a definição de quais clientes são atendidos em cada parada do veículo, os quais geralmente são considerados como dados de pré-processamento na literatura. Desigualdades válidas e esquemas de decomposição de Benders foram discutidos para desenvolver algoritmos de solução exatos para essas novas variantes do PRVJTME. Experimentos mostram a importância de se aplicar essas variantes em vez da abordagem comum na literatura, levando a economias de custo da ordem de 10%.

Palavras-chave: roteamento de veículos; entrega de última milha; problema de localização-roteamento em dois níveis; múltiplos entregadores; decomposição de Benders.

Contents

1 Introduction **13**

2 The two-echelon location-routing problem: A comparative analysis of novel and existing compact formulations **19**

2.1 Introduction 20
2.2 Literature review 21
2.3 Problem definition and mathematical formulations 23
2.3.1 Formulation with vehicle index variables (CF1) 25
2.3.2 Formulation with two-index arc variables and binary assignments (CF2) 28
2.3.3 Formulation with two-index arc variables and continuous assignments (CF3) 31
2.4 Comparison of LRs 32
2.5 Computational experiments 35
2.5.1 Comparison of CFs 36
2.5.2 The impact of multiple platforms 41
2.5.3 Experiments with VIs 43
2.6 Conclusion 48

3 An exact method for a last-mile delivery routing problem with multiple delivery-men **50**

3.1 Introduction 50
3.2 Literature review 52
3.3 Problem definition 54
3.4 Mathematical formulation 56
3.4.1 Valid inequalities 58
3.5 Benders decomposition 59
3.5.1 Master problem 59
3.5.2 Subproblem 62
3.5.3 Lower bounds 63
3.5.4 Branch-and-Benders-cut 65
3.6 Computational experiments 65

3.6.1 Instances . . . 66
3.6.2 Compact formulation and valid inequalities . . . 66
3.6.3 Branch-and-Benders-cut algorithm . . . 69
3.6.4 Managerial insights . . . 72
3.7 Conclusion . . . 76

4 Last-mile delivery with multiple deliverymen: formulation and exact solution methods for a rich vehicle routing problem 78
4.1 Introduction . . . 79
4.2 Literature review . . . 80
4.3 Problem definition . . . 82
4.4 Mathematical formulation . . . 83
4.4.1 Theoretical properties . . . 86
4.4.2 Valid inequalities . . . 88
4.5 Benders decomposition . . . 90
4.5.1 Master Problem . . . 90
4.5.2 Subproblem . . . 92
4.5.3 Branch-and-Benders-cut algorithm . . . 94
4.5.4 Improvements . . . 94
4.5.5 MIP heuristic . . . 96
4.6 Computational experiments . . . 96
4.6.1 Instances . . . 97
4.6.2 Compact formulation . . . 98
4.6.3 Branch-and-Benders-cut . . . 101
4.6.4 Managerial insights . . . 104
4.7 Conclusion . . . 107

5 Conclusion 109

6 References 111

List of Tables

2.1 A summary of the main formulations found for the 2E-LRP in the literature. . . 23
2.2 Results of the MIP solver for different CFs. 37
2.3 The impact of the number of potential platforms on the CFs performances. . . . 43
2.4 Different VI configurations for each formulation. 44
2.5 The impact of including VIs in CF1 for small instances (71 instances). 45
2.6 The impact of including VIs in ICF1 for small instances (71 instances). 45
2.7 The impact of including VIs in ICF2 for small and medium instances. 46
2.8 The impact of including VIs in CF3. 47

3.1 Results of the experiments with CF and different sets of VIs. 67
3.2 Impact of cut improvements in the BBC method. 70
3.3 Results of the experiments with the best versions of the CF and BBC approaches. 70
3.4 Average number of cuts and separation times in the BBC algorithm. 72
3.5 The importance of considering deliveryman routes. 73
3.6 Costs sensitivity analysis. 74
3.7 Further advantages of multiple deliverymen. 74
3.8 Solution variation after reclustering. 76

4.1 Results of the CF configurations for instances of size 5–20. 99
4.2 Results of the CF configurations for instances of size 10–40. 100
4.3 Results of the BBCs for instances of size 5–20. 102
4.4 Results of the BBCs for instances of size 10–40. 103
4.5 The impact of different VRPTWMD variants on the solution quality. 105

List of Figures

1.1 The two-echelon location-routing problem. 14
1.2 The vehicle routing problem with time windows, multiple deliverymen, and two-level routing. 16
1.3 The vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing. 16

2.1 An illustrative example of the 2E-LRP. 24
2.2 An example for proving that ICF1 does not have a stronger LR than CF2. . . . 33
2.3 A visual representation of the relationships between different 2E-LRP formulations. 35
2.4 Results of the MIP solver for different CFs and sizes. 38
2.5 Average number of constraints, variables, and binary variables for different CFs and sizes. 39
2.6 Performance profile of the MIP solver for different CFs and sizes considering the UB. 41
2.7 Performance profile of the MIP solver for different CFs and sizes considering the optimality gap. 42

3.1 An illustrative example of the VRPTWMD2R. 55
3.2 A trade-off between deliveryman routes cost and time. 55
3.3 Different solutions by minimizing deliveryman routes cost or time. 64
3.4 Convergence curves for instance R110 with size 25–125. 68
3.5 Convergence of the BBC2 method for instance R110 with size 25–125. 71

4.1 An illustrative example of the VRPTWMDC2R. 83

1 Introduction

In operational contexts of delivery operations, a very important question is: “which are the routes that the vehicles should follow to serve the customers?”. In this decision, several factors should be taken into account, such as customer information (e.g., demand and time windows), vehicle characteristics (e.g., capacity and speed), and business strategy (e.g., service level).

In 1959, Dantzig and Ramser proposed what was later called the vehicle routing problem (VRP), which is an optimization problem that aims at minimizing the costs of the vehicle routes, while serving all customers (Dantzig; Ramser, 1959). Since then, many variants of the VRP were proposed to reflect the characteristics of different business models. The most traditional variant is the capacitated VRP (Queiroga; Sadykov; Uchoa, 2021), which assumes that vehicles have load capacity. There is also the VRP with time windows (Toth; Vigo, 2014), in which customers have time windows in which the service must occur; the pickup-and-delivery routing problem (Furtado; Munari; Morabito, 2017), in which the goods must be collected and distributed; the split-delivery VRP (Munari; Savelsbergh, 2022), that assumes that the demand of customers can be served by more than one vehicle; among many others.

A particularly interesting area of application of VRP variants is the last-mile delivery, corresponding to the delivery to the final customers. Last-mile delivery systems are more and more complex nowadays due to the increasing demand for efficient deliveries in densely populated urban centers. A very common approach among companies is the adoption of some sort of two-echelon system (Cuda; Guastaroba; Speranza, 2015; Li et al., 2021a; Sluijk et al., 2023). In these schemes, larger vehicles travel from the depots to intermediary points from which smaller vehicles take the goods to the final customers. These intermediary points can be either transshipment facilities (Li et al., 2021b; Escobar-Vargas; Crainic, 2024), or parking locations from which smaller vehicles such as drones and robots (Moshref-Javadi; Winkenbach, 2021; Alfandari; Ljubić; De Melo da Silva, 2022), or walking deliverymen (Pureza; Morabito; Reimann, 2012; Cabrera; Cordeau; Mendoza, 2022) take goods from the vehicles to the customers. A crucial question in these delivery systems is the definition of the routes to be traveled by the vehicles.

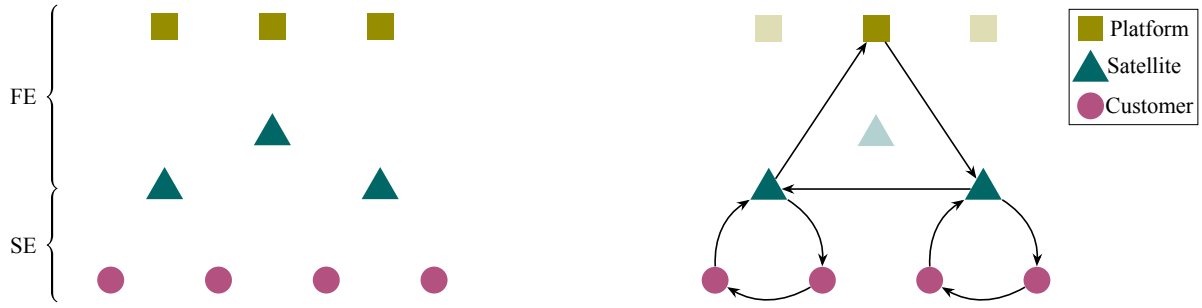
In this dissertation, we study two combinatorial optimization problems related to two-echelon routing in last-mile delivery: the two-echelon location-routing problem (2E-LRP) and the vehicle routing problem with tiwe windows and multiple deliverymen (VRPTWMD).

The first problem studied, the 2E-LRP, was originally proposed by Boccia et al. (2011). In

this problem, there are two sets of facilities: the platforms, which are larger and where the goods are stored (equivalent to depots), and the satellites, which are smaller and where the goods are transshipped. There is also a set of customers with demands. Larger vehicles take goods from the platforms to the satellites, corresponding to the first-echelon (FE); and smaller vehicles take these goods to the customers in the second-echelon (SE). The aim of the problem is to define which facilities to open (or use) and which routes should the FE and SE vehicles perform, while minimizing the costs – fixed costs associated with opening the facilities and using the vehicles, and distance costs associated with the vehicle routes.

The 2E-LRP reflects applications in which there are recurrent deliveries in urban areas where the circulation of large vehicles is limited or the traffic significantly slows the delivery process. This way, the larger vehicles are used in the FE, where there are no circulation restrictions, and the smaller vehicles are used in the SE, since they are allowed to transit for being small. Moreover, where there is an excessive traffic, it is pointless to use large vehicles, since, in these contexts, the routes are often limited by their duration, not the capacity of the vehicle. As smaller vehicles are cheaper, the business scheme proposed by the 2E-LRP leads to more cost-efficient deliveries.

Figure 1.1 illustrates the 2E-LRP dynamics. Figure 1.1a presents an instance with three potential platforms, three potential satellites, and four customers. Figure 1.1b portrays a solution for this instance, in which only one platform and two satellites are opened (the shaded ones are not used). One FE vehicle visits both satellites and two SE vehicles serve the customers.



(a) Instance. (b) Solution.

Figure 1.1: The two-echelon location-routing problem.

Most of the literature on this problem is based on mixed-integer programming (MIP) formulations that have variables with vehicle index. However, experiments have shown that these formulations are not effective to solve the problem. The main contribution of this dissertation towards the 2E-LRP is the proposition of two novel compact formulations (with a polynomial number of variables and constraints) that do not require the vehicle index in the variables. The quality of their linear programming relaxations is theoretically compared with the benchmark formulation, and extensive computational experiments are performed to evaluate the performance of a commercial MIP solver upon solving instances of the 2E-LRP with each of the formulations. The obtained results indicate that the novel formulations outperform the benchmark

one.

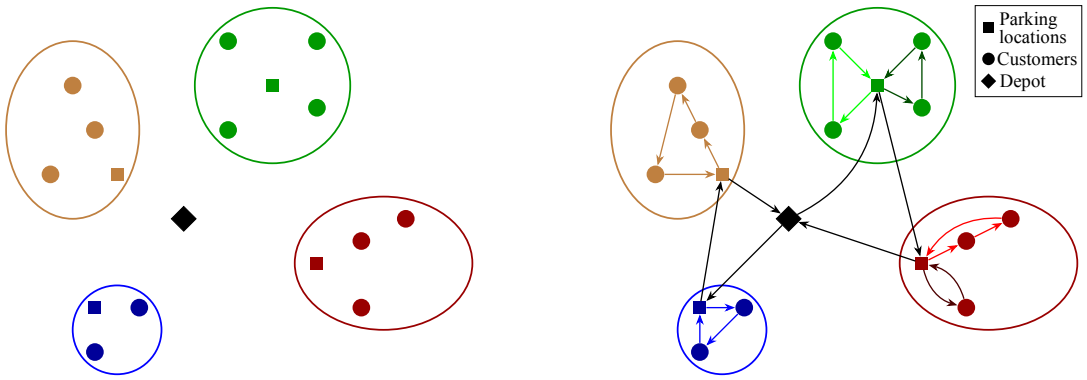
The second problem studied, the VRPTWMD, was introduced by Pureza, Morabito, and Reimann (2012). It reflects a delivery system in which each vehicle may travel with more than one deliveryman to increase delivery efficiency. This way, every time the vehicle stops, many customers (cluster) are served in parallel by the multiple deliverymen. This allows each vehicle to serve more customers than it would in a traditional vehicle routing problem with time windows, reducing costs.

When defining the problem, Pureza, Morabito, and Reimann (2012) introduced two simplifying hypotheses to improve its tractability: (*i*) the definition of which customers are to be served by a single vehicle stop (clusters) can be predefined, and (*ii*) the deliveryman routes can be approximated in a preprocessing phase. The majority of the literature that followed worked with these hypotheses (Álvarez; Munari, 2017; Munari; Morabito, 2018; De La Vega; Munari; Morabito, 2020). Senarclens de Grancy and Reimann (2015) and Senarclens de Grancy (2015) extended the problem by including the customer clustering in the optimization, while still considering that the deliveryman routes can be approximated. To the best of our knowledge, there is no work that solves the VRPTWMD considering the optimization of the deliveryman routes.

The main contribution of this dissertation regarding the VRPTWMD is to bridge this gap. We introduce two novel and realistic variants of the VRPTWMD: one that includes the deliveryman routes in the optimization while still considering that the clusters are predefined, and another that optimizes both these routes and the customer clustering.

The first variant introduced is the VRPTWMD with two-level routing (VRPTWMD2R), which includes the decisions of which deliveryman routes to perform in the optimization, while still considering that the customer clusters are predefined. Figure 1.2 represents this problem. In Figure 1.2a, an instance of the problem is illustrated, with customers divided in clusters. Each cluster has a parking location and a few customers. The VRPTWMD2R aims at defining the vehicle and deliveryman routes that serve the customers while minimizing the costs – fixed costs of vehicles and deliveryman, and distance costs associated with vehicle and deliveryman routes. Figure 1.2b portrays a feasible solution for this instance. In the right-hand side of the picture, one vehicle with two deliverymen serve the upper-right green and lower-right red clusters, while, in the left-hand side, another vehicle with only one deliveryman serves the remaining ones.

The second variant studied is the VRPTWMD with clustering and two-level routing (VRPTWMDC2R). In this variant, on top of considering the deliveryman routes, the customer clustering is also included in the optimization. This is illustrated in Figure 1.3. The differences between the instance portrayed in Figure 1.3a and the one in 1.2a clearly show that, unlike in the VRPTWMD2R, in the VRPTWMDC2R there are not predefined clusters and, instead, there are a set of parking locations that may be used and a set of customers. It is part of the problem the definition of which parking locations to use and which customers to serve from each parking location, as shown in Figure 1.3b. This figure (1.3b) represents a solution for this instance, with the proper definition of customer clusters, the decision of which parking locations to visit, and

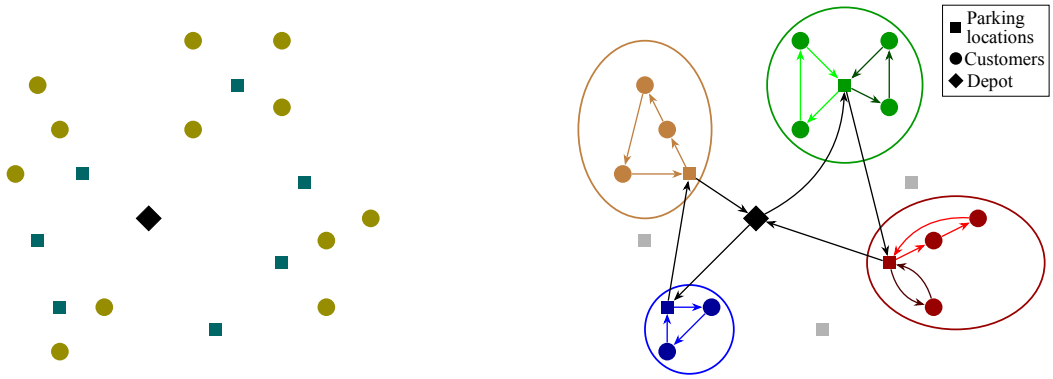


(a) Instance. (b) Solution.

Figure 1.2: The vehicle routing problem with time windows, multiple deliverymen, and two-level routing.

the vehicle and deliveryman routes.

In summary, the main objectives of this dissertation are (*i*) to propose more efficient compact formulations for the 2E-LRP, and (*ii*) to include the deliveryman routes and the customer clustering in the VRPTWMD. To achieve this, the following specific objectives must be pursued: (*i*) to review the pertinent literature on the 2E-LRP and the VRPTWMD, (*ii*) to propose novel formulations and valid inequalities for the 2E-LRP, (*iii*) to introduce two realistic extensions for the VRPTWMD and formulate them, (*iv*) to propose valid inequalities and exact methods for these variants, and (*v*) to discuss the benefits of considering these variants compared to the literature approach to the VRPTWMD.



(a) Instance. (b) Solution.

Figure 1.3: The vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing.

This dissertation is organized as a collection of three papers, in which all research developments and results are detailed. The first paper is presented in Chapter 2, entitled *The two-echelon location-routing problem: A comparative analysis of novel and existing compact formulations* and coauthored with Prof. Leandro C. Coelho (from Université Laval, in Canada), Prof. Reinaldo Morabito, and Prof. Pedro Munari (both from the Federal University of São

Carlos, in Brazil). It provides a brief literature review of the 2E-LRP and introduces two novel compact formulations for the 2E-LRP. The new formulations are compared with the benchmark one, showing that they greatly outperform the benchmark. Moreover, the best known lower bounds are improved for 125 out of the 131 benchmark instances evaluated and 55 new optimal solutions are found. This paper is publicly available at the CIRRELT repository (Senna et al., 2024b).

Chapter 3 contains the second paper, entitled *An exact method for a last-mile delivery routing problem with multiple deliverymen* and coauthored with Prof. Leandro C. Coelho (from Université Laval, in Canada), Prof. Reinaldo Morabito, and Prof. Pedro Munari (both from the Federal University of São Carlos, in Brazil). This paper surveys the literature on the VRPTWMD, and introduces a variant that includes the deliveryman routes in the optimization problem (VRPTWMD2R). This variant is formulated, valid inequalities are proposed, and it is decomposed in a Benders (1962) fashion to solve it exactly in a branch-and-Benders-cut (BBC) scheme (Moreno; Munari; Alem, 2019, 2020). The obtained results show the efficiency of the proposed methodology, solving realistic sized instances within reasonable times. In addition, the paper empirically shows the importance of considering the deliveryman routes from a costs minimization perspective, evaluates the impact of the deliveryman usage on the solution quality, and discusses opportunities for reducing costs and greenhouse gases emissions with the clever adoption of multiple deliverymen. This paper has been published at the *European Journal of Operational Research* (Senna et al., 2024a).

The third paper extends the VRPTWMD by considering both the deliveryman routes and the customer clustering. It is entitled *Last-mile delivery with multiple deliverymen: formulation and exact solution methods for a rich vehicle routing problem*, and is presented in Chapter 4. It is coauthored with Prof. Leandro C. Coelho (from Université Laval, in Canada), Prof. Reinaldo Morabito, and Prof. Pedro Munari (both from the Federal University of São Carlos, in Brazil). This paper introduces the VRPTWMDC2R, presents a mathematical formulation, discusses some theoretical properties, and proposes useful lower bounds. These theoretical results are used to develop valid inequalities that, in practice, significantly improve the MIP solver performance. The VRPTWMDC2R is decomposed in a Benders fashion and a branch-and-Benders-cut is proposed to solve it (Moreno; Munari; Alem, 2019, 2020). Managerial insights show the importance of including the deliveryman routes and the customer clustering in the optimization problem.

Concisely, the main contributions of this dissertation are new approaches for well-known routing problems applied to last-mile delivery. Specifically, the contributions concerning the 2E-LRP are (*i*) the introduction of two novel formulations based on two-index arc variables, (*ii*) the proposition of valid inequalities for these formulations, and (*iii*) the discovery of 125 new lower bounds and 55 new optimal solutions for benchmark instances. For the VRPTWMD, the contributions are (*iv*) the introduction of a realistic variant including the deliveryman routes in the optimization, (*v*) the introduction of another realistic variant that encompasses both the deliv-

eryman routes and the customer clustering in the optimization, (*vi*) MIP formulations and valid inequalities for both variants, (*vii*) Benders decomposition schemes and branch-and-Benders-cut algorithms to solve these variants, and (*viii*) computational experiments to evaluate the performance of the proposed approaches and the impact of the new variants on the solution quality.

Since this document is organized as a collection of papers, Chapters 2 to 4 should be treated as independent documents. Specially considering notation, even though there are many similarities, the notation of each paper should be considered independently. Throughout the document, the terms “two-level” and “two-echelon” are used interchangeably, with the choice of which expression to use being defined by the literature over which the corresponding paper is based. The remainder of this document is structured as follows: Chapters 2 to 4 present the three papers that compose this dissertation, and Chapter 5 presents concluding remarks.

2 The two-echelon location-routing problem: A comparative analysis of novel and existing compact formulations

Abstract

The two-echelon location-routing problem (2E-LRP) is a well-known problem in the literature that is commonly used to address applications in which deliveries occur at two levels. It concerns the location of facilities and the routing of vehicle fleets. Most studies addressing this problem and its variants rely on mixed-integer programming (MIP) formulations that are compact (i.e., have a polynomial number of variables and constraints). Although the formulations with two-index arc variables tend to perform better than those with vehicle index variables in vehicle routing problems, most of the literature on the 2E-LRP is based on the latter. In this paper, we present a comparative analysis of three compact formulations for the 2E-LRP: a literature-based formulation with vehicle index variables, and two novel formulations with two-index arc variables. Additionally, we propose enhancements for the literature-based formulation and polynomial valid inequalities for all of them. The linear programming relaxations of these formulations are compared, showing that those of the two-index formulations are stronger. Extensive computational experiments evaluate the formulations' performances on a general-purpose MIP solver. The results show that the formulations with vehicle index variables, despite being the standard approach in the literature, lead to poor solver performance, failing to find feasible solutions even for instances with only 50 customers. In fact, the best performance comes from the novel formulations, one of which leads to feasible solutions for all benchmark instances evaluated. Valid inequalities can be used to improve this performance even further. These experiments resulted in the discovery of 125 new best known lower bounds and 55 new optimal solutions (out of 131 benchmark instances evaluated).[1]

[1] This chapter is a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos). It is publicly available at the CIRRELT repository (Senna et al., 2024b).

2.1 Introduction

The continuous worsening of traffic conditions in urban centers has prompted many municipalities to impose restrictions on the traffic of large vehicles in the cities (Enthoven et al., 2020; Friedrich; Elbert, 2022). This, associated with a growing demand for urban deliveries, has led many companies to develop new logistics schemes. In city logistics, one particularly popular approach is the delivery in two echelons (Cuda; Guastaroba; Speranza, 2015; Senna et al., 2024a). This way, larger vehicles transport goods from central depots (platforms) to smaller facilities (satellites) closer to the urban centers, in the first echelon (FE). From these satellites, smaller vehicles serve the customers in the second echelon (SE). Thus, the long distances are traveled by more cost-efficient vehicles while complying with constraints on urban traffic.

From an operational standpoint, it is important to determine the most efficient vehicle routes, which is the concern of the well-known two-echelon vehicle routing problem (2E-VRP), as reviewed by Sluijk et al. (2023). From a strategic and tactical perspective, one must consider both location and routing decisions, which leads to the so-called location-routing problems (Prodhon; Prins, 2014). In particular, the two-echelon location-routing problem (2E-LRP) studies the decisions regarding the opening of facilities (platforms and satellites) and the routes of FE and SE vehicles, with the objective of minimizing overall costs (Drexl; Schneider, 2015).

The idea of integrating location and routing decisions in a two-echelon scheme can be traced back to the works of Jacobsen and Madsen (1980) and Madsen (1983). However, it was only in 2011 that Boccia et al. formally defined and formulated the 2E-LRP (Boccia et al., 2011). The authors introduced three mixed-integer programming (MIP) formulations, two of which were compact (i.e., with polynomial numbers of variables and constraints) and one was extensive (i.e., with an exponential number of variables). The results of computational experiments indicated that the best compact formulation (CF) was based on arc variables with a vehicle index, clearly outperforming the one based on two-index arc variables. Results for the extensive formulation were not presented. Since then, most papers dealing with CFs for the 2E-LRP and its variants have relied on this vehicle index-based formulation.

The main difficulty in designing two-index arc variables CFs for the 2E-LRP is ensuring that the vehicles return to the facility they left from. In formulations with a vehicle index, this is simply made by flow conservation constraints, which ensure that the vehicle flow arriving at a node in one vehicle must leave this node in the same vehicle. Hence, the vehicles must make closed loops. In formulations without the vehicle index, this constraint does not work anymore in this sense because, although the vehicle flow should be maintained, it is possible that the vehicle arriving at a facility is not the same that left it. Thus, additional variables and constraints are required to guarantee that a vehicle starts and ends its route at the same facility.

In this paper, we propose two novel CFs with two-index arc variables for the 2E-LRP. The main difference between them is exactly the variables and constraints used to ensure that vehicles return to the facilities they departed from. Both of these formulations outperform the original

formulation with vehicle index variables in general-purpose MIP solvers. Because the majority of papers addressing the 2E-LRP rely on CFs, this may be a significant development, as it will allow future researchers and practitioners to work with simple yet more powerful options. The contributions of this paper are fivefold:

- A vehicle index-based formulation adapted from Boccia et al. (2011) by revising some minor inaccuracies and two novel formulations based on two-index arc variables;
- New valid inequalities for all of the proposed formulations;
- A theoretical comparison of the linear programming relaxations (LRs) of the different formulations;
- Extensive computational experiments to assess which is the best CF for the 2E-LRP when relying on a general-purpose MIP solver;
- 125 best known lower bounds for benchmark instances and 55 new optimal solutions (out of 131 instances evaluated).

The remainder of this paper is organized as follows. Section 2.2 provides an overview of the literature on the 2E-LRP. In Section 2.3, we formally define the problem, present the different formulations, and introduce the valid inequalities. Section 2.4 provides a theoretical comparison of the LRs of the formulations. In Section 2.5, we discuss the results of the computational experiments. Finally, Section 2.6 presents concluding remarks.

2.2 Literature review

This section presents a review of the literature on the 2E-LRP and its variants, with a particular focus on the formulations used in these publications. We restrict our review to papers that present MIP formulations. For comprehensive reviews, we refer the reader to the works of Prodhon and Prins (2014), Cuda, Guastaroba, and Speranza (2015), and Drexl and Schneider (2015).

Boccia et al. (2011) were the first to formally define and formulate the 2E-LRP. They presented three different MIP formulations. The first is a CF that considers binary arc variables with a vehicle index (three-index formulation). The second one is also compact and avoids the vehicle index by using only two-index arc variables. The third one is an extensive formulation with an exponential number of variables representing all the feasible routes for the problem. Their computational experiments only provide results for the CFs and demonstrate empirically that the formulation with vehicle index variables is better than the alternative, which had a significant impact on subsequent literature.

Nguyen, Prins, and Prodhon (2012a) were the first to develop metaheuristics for the 2E-LRP, while also presenting the formulation with vehicle index variables introduced by Boccia et al.

(2011). Contardo, Hemmelmayr, and Crainic (2012) and Nguyen, Prins, and Prodhon (2012b) also worked on the 2E-LRP as defined by Boccia et al. (2011) by proposing metaheuristics and extensive two-index arc variables-based formulations with an exponential number of constraints. Govindan et al. (2014) extended the problem to encompass time windows in a multi-objective approach to design a sustainable perishable food supply chain. They presented a CF based on the three-index formulation proposed by Boccia et al. (2011). Breunig et al. (2016) extended the problem by considering split deliveries in the FE and provided an extensive formulation with an exponential number of variables.

Rahmani, Cherif-Khettaf, and Oulamara (2016) and Wang et al. (2018) adapted the 2E-LRP to two beverage distribution applications, also presenting CFs based on arc variables with a vehicle index. Pichka et al. (2018) extended the problem for an open routing situation and Zhao, Wang, and Souza (2017) looked at the particularities of heterogeneous fleets. Both papers propose CFs based on variables with a vehicle index. Darvish et al. (2019) incorporated the notion of flexibility into the 2E-LRP, modeling it with a CF and presenting valid inequalities and an exact method. Dai et al. (2019) addressed the 2E-LRP as well as two other extensions considering three and four echelons, modeling them with variables with a vehicle index.

The 2E-LRP has also been applied to model off-shore oil and gas supply chains (Amiri; Amin; Tavakkoli-Moghaddam, 2019), postal services (Mirhedayatian et al., 2021), electric vehicles applications (Wang; Miao; Zhang, 2021), disaster waste clean-up in humanitarian contexts (Cheng et al., 2022), cold supply chains (Wang et al., 2023), and other city logistics situations (Agnimo et al., 2023). Sutrisno and Yang (2023) looked at the problem with mobile satellites instead of fixed ones, and Escobar-Vargas and Crainic (2024) dealt with synchronization constraints. All of them used variables with a vehicle index.

Yıldız, Karaoğlan, and Altiparmak (2023) discussed a variant of the 2E-LRP with pickup and delivery. They relied on a formulation with two-index arc variables based on assignment variables used for the 2E-VRP (Belgin; Karaoğlan; Altiparmak, 2018). They adapted this formulation to the 2E-LRP, but without including the platforms' capacity constraints since the way it was modeled would create a non-linearity. In Section 2.3.2, we introduce a formulation that is based on what they proposed while including these capacity constraints by adopting a commodity flow-based formulation. We also present an improvement to this formulation.

Tian and Hu (2023) and Ben Mohamed et al. (2023) proposed branch-and-price algorithms, considering extensive formulations with an exponential number of variables. The first study considered a variant of the 2E-LRP with satellite recommendations whereas the second analyzed a multi-period stochastic variant.

Table 2.1 provides a summary of the information presented, analyzing the formulation characteristics of each work. Of the 23 works presented, 18 of them (78%) defined their problems with CFs. Of those, 16 (89%) had vehicle index variables in their formulations. Moreover, Amiri, Amin, and Tavakkoli-Moghaddam (2019) worked with vehicle index variables despite having an extensive formulation. Only four of the works presented two-index arc variables. Of

those, only two presented CFs and Boccia et al. (2011) showed that their two-index variables formulation performed worse than the vehicle index one, while Yıldız, Karaoğlan, and Altiparmak (2023) ignored the platforms capacity constraints. It is important to note that, in the context of this discussion, we do not refer to echelon-related indices because, in many works, the FE and SE arc variables have different notations.

Reference	Compact formulation	Vehicle index variables	Two-index arc variables
Boccia et al. (2011)	✓	✓	✓
Contardo, Hemmelmayr, and Crainic (2012)			✓
Nguyen, Prins, and Prodhon (2012a)	✓	✓	
Nguyen, Prins, and Prodhon (2012b)			✓
Govindan et al. (2014)	✓	✓	
Breunig et al. (2016)			
Rahmani, Cherif-Khettaf, and Oulamara (2016)	✓	✓	
Zhao, Wang, and Souza (2017)	✓	✓	
Pichka et al. (2018)	✓	✓	
Wang et al. (2018)	✓	✓	
Darvish et al. (2019)	✓		
Dai et al. (2019)	✓	✓	
Amiri, Amin, and Tavakkoli-Moghaddam (2019)		✓	
Mirhedayatian et al. (2021)	✓	✓	
Wang, Miao, and Zhang (2021)	✓	✓	
Cheng et al. (2022)	✓	✓	
Wang et al. (2023)	✓	✓	
Agnimo et al. (2023)	✓	✓	
Tian and Hu (2023)	✓	✓	
Ben Mohamed et al. (2023)			
Yıldız, Karaoğlan, and Altiparmak (2023)	✓		✓
Sutrisno and Yang (2023)	✓	✓	
Escobar-Vargas and Crainic (2024)	✓	✓	

Table 2.1: A summary of the main formulations found for the 2E-LRP in the literature.

This outcome indicates the importance of CFs for the 2E-LRP because, even though most of the papers present tailored optimization methods (exact and heuristic) for their problems, they usually apply CFs to formally define the addressed variants and compare the performance of their methods with that of the CF. Hence, the better the formulation, the fairer the comparison. Moreover, given that the vast majority of formulations include vehicle index variables, it is important to assess whether this is the best approach. The present paper aims at solving this issue by presenting novel formulations without vehicle index variables and by comparing all of them theoretically and computationally.

2.3 Problem definition and mathematical formulations

The 2E-LRP is defined over a graph $G = (\mathcal{N}, \mathcal{A})$. The node set is $\mathcal{N} = \mathcal{P} \cup \mathcal{S} \cup \mathcal{C}$, with $\mathcal{P}$ being the set of potential platforms, $\mathcal{S}$ the set of potential satellites, and $\mathcal{C}$ the set of customers. Platforms and satellites are also called facilities. The set of echelons is $\mathcal{E} = \{1, 2\}$, with $e = 1$

representing the FE and $e = 2$ the SE. We define sets $\mathcal{N}^1 = \mathcal{P} \cup \mathcal{S}$ and $\mathcal{N}^2 = \mathcal{S} \cup \mathcal{C}$. To improve notation, we shall denote by $\mathcal{O}^e$ and $\mathcal{D}^e$ the sets of origins and destinations in echelon e, i.e., $\mathcal{O}^1 = \mathcal{P}$, $\mathcal{D}^1 = \mathcal{S}$, $\mathcal{O}^2 = \mathcal{S}$, and $\mathcal{D}^2 = \mathcal{C}$. The set of arcs is $\mathcal{A} = \mathcal{A}^1 \cup \mathcal{A}^2$, with $\mathcal{A}^1 = \{(i,j)|(i \in \mathcal{P}, j \in \mathcal{S}) \vee (i \in \mathcal{S}, j \in \mathcal{P}) \vee (i \in \mathcal{S}, j \in \mathcal{S}, i \neq j)\}$ and $\mathcal{A}^2 = \{(i,j)|(i \in \mathcal{S}, j \in \mathcal{C}) \vee (i \in \mathcal{C}, j \in \mathcal{S}) \vee (i \in \mathcal{C}, j \in \mathcal{C}, i \neq j)\}$.

In each echelon, there is an unlimited and homogeneous fleet (FE and SE vehicles may be different). FE vehicles take goods from the platforms to the satellites, where they are transshipped and delivered to the customers by the SE vehicles. Every vehicle route starts and ends at the same facility. In the 2E-LRP, each facility $i \in \mathcal{N}^1$ has a fixed cost H_i associated with opening it and a capacity B_i. FE and SE vehicles have capacities Q^1 and Q^2 and fixed costs f^1 and f^2, respectively. Each customer $i \in \mathcal{C}$ has a demand q_i. The cost of traveling in arc $(i,j) \in \mathcal{A}^e$ in echelon $e \in \mathcal{E}$ is c^e_{ij}. It is worth noting that superindex e in c^e_{ij} could be suppressed, since each arc only belongs to one echelon. However, we opted to keep it since it makes notation clearer both in this parameter and in some variables.

The goal of the 2E-LRP is to determine the optimal subset of facilities to open, along with the least-cost FE and SE routes that can serve all customers. Figure 2.1 illustrates the problem by presenting a feasible solution to an instance with three potential platforms, three potential satellites, and four customers. This solution uses a single platform and two satellites (the shaded ones are potential facilities that are not selected in this solution). In the FE, the vehicle serves the two satellites from a single platform, whereas in the SE, two vehicles serve the customers.

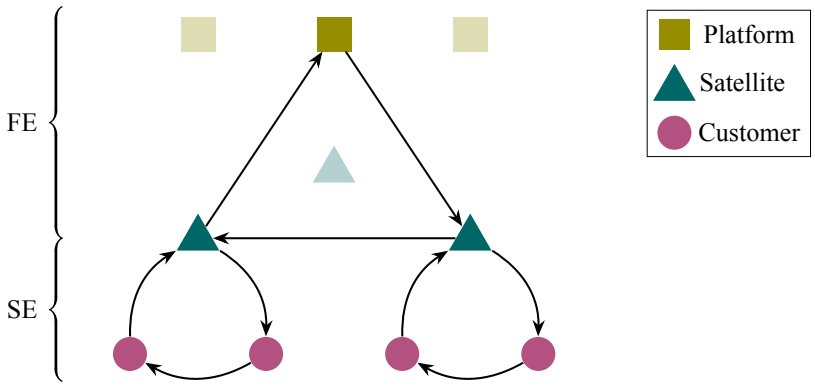


Figure 2.1: An illustrative example of the 2E-LRP.

We present three CFs for this problem. Section 2.3.1 presents a formulation with vehicle index variables proposed for the 2E-LRP (CF1) based on the one introduced by Boccia et al. (2011), and discusses an improvement of CF1 by considering commodity flow constraints (ICF1). We do not present the other formulations proposed by Boccia et al. (2011) since their experiments proved that these formulations performed worse. Section 2.3.2 introduces a formulation with two-index arc variables based on binary assignment variables (CF2), adapted from what is proposed by Yıldız, Karaoğlan, and Altiparmak (2023), and a possible enhancement (ICF2). In Section 2.3.3, another formulation with two-index arc variables is presented, without

binary assignment variables (CF3). Valid inequalities (VIs) are discussed for all formulations. As discussed in Section 2.1, the main difference between the two formulations with two-index arc variables (CF2 and CF3) is the variables and constraints that are used to ensure that each vehicle returns to the facility it left from. In what follows, the binary variable y_i is common to all proposed formulations and indicates whether a facility $i \in \mathcal{N}^1$ is opened.

2.3.1 Formulation with vehicle index variables (CF1)

In this section, we present the formulation with three-index variables for the 2E-LRP as introduced by Boccia et al. (2011), but we fix minor errors of their presentation. This formulation requires additional sets $\mathcal{K}^e$ of vehicles in echelon $e \in \mathcal{E}$.

The binary variable w^2_{si} indicates whether customer $i \in \mathcal{C}$ is assigned to satellite $s \in \mathcal{S}$. Also, the binary variable x^e_{ijk} indicates whether a vehicle $k \in \mathcal{K}^e$ travels through arc $(i,j) \in \mathcal{A}^e$ in echelon $e \in \mathcal{E}$. Another binary variable z_k is required to indicate whether vehicle $k \in \mathcal{K}^1 \cup \mathcal{K}^2$ is used. Load flow from platform $p \in \mathcal{P}$ to satellite $s \in \mathcal{S}$ in vehicle $k \in \mathcal{K}^1$ is controlled by the continuous and non-negative variable g_{psk}. Finally, u^e_i is an auxiliary variable for subtour elimination that indicates the position of node $i \in \mathcal{N}^e$ in a route in echelon $e \in \mathcal{E}$.

The formulation introduced by Boccia et al. (2011) for the 2E-LRP is:

$$\text{(CF1) min} \sum_{i\in\mathcal{N}^1} H_i y_i + \sum_{e\in\mathcal{E}}\sum_{k\in\mathcal{K}^e} f^e z_k + \sum_{e\in\mathcal{E}}\sum_{k\in\mathcal{K}^e}\sum_{(i,j)\in\mathcal{A}^e} c^e_{ij} x^e_{ijk} \tag{2.1}$$

$$\text{s.t.} \sum_{i:(i,j)\in\mathcal{A}^e} x^e_{ijk} = \sum_{i:(j,i)\in\mathcal{A}^e} x^e_{jik},\ \forall\, j \in \mathcal{N}^e, k \in \mathcal{K}^e, e \in \mathcal{E} \tag{2.2}$$

$$u^e_j \geq u^e_i + 1 - |\mathcal{D}^e|\left(1 - \sum_{k\in\mathcal{K}^e} x^e_{ijk}\right),\ \forall\,(i,j) \in \mathcal{A}^e, e \in \mathcal{E} \tag{2.3}$$

$$\sum_{j\in\mathcal{O}^e}\sum_{i:(i,j)\in\mathcal{A}^e} x^e_{ijk} \leq 1,\ \forall\, k \in \mathcal{K}^e, e \in \mathcal{E} \tag{2.4}$$

$$\sum_{k\in\mathcal{K}^1}\sum_{j:(s,j)\in\mathcal{A}^1} x^1_{sjk} = y_s,\ \forall\, s \in \mathcal{S} \tag{2.5}$$

$$\sum_{k\in\mathcal{K}^2}\sum_{j:(i,j)\in\mathcal{A}^2} x^2_{ijk} = 1,\ \forall\, i \in \mathcal{C} \tag{2.6}$$

$$\sum_{s\in S} w^2_{si} = 1,\ \forall\, i \in \mathcal{C} \tag{2.7}$$

$$\sum_{j:(i,j)\in\mathcal{A}^2} x^2_{ijk} + \sum_{j:(s,j)\in\mathcal{A}^2} x^2_{sjk} - w^2_{si} \leq 1,\ \forall\, i \in \mathcal{C}, s \in \mathcal{S}, k \in \mathcal{K}^2 \tag{2.8}$$

$$\sum_{k\in\mathcal{K}^1}\sum_{p\in\mathcal{P}} g_{psk} = \sum_{i\in\mathcal{C}} q_i w^2_{si},\ \forall\, s \in \mathcal{S} \tag{2.9}$$

$$\sum_{k\in\mathcal{K}^1}\sum_{s\in\mathcal{S}} g_{psk} \leq B_p y_p,\ \forall\, p \in \mathcal{P} \tag{2.10}$$

$$\sum_{k\in\mathcal{K}^1}\sum_{p\in\mathcal{P}} g_{psk} \leq B_s y_s,\ \forall\, s \in \mathcal{S} \tag{2.11}$$

$$Q^1 \sum_{j:(s,j)\in\mathcal{A}^1} x^1_{sjk} \geq g_{psk},\ \forall\, p \in \mathcal{P}, s \in \mathcal{S}, k \in \mathcal{K}^1 \tag{2.12}$$

$$Q^1 \sum_{j:(p,j)\in\mathcal{A}^1} x^1_{pjk} \geq g_{psk},\ \forall\, p \in \mathcal{P}, s \in \mathcal{S}, k \in \mathcal{K}^1 \tag{2.13}$$

$$\sum_{p\in\mathcal{P}} \sum_{s\in\mathcal{S}} g_{psk} \leq Q^1 z_k,\ \forall\, k \in \mathcal{K}^1 \tag{2.14}$$

$$\sum_{i\in\mathcal{C}} \sum_{j:(i,j)\in\mathcal{A}^2} q_i x^2_{ijk} \leq Q^2 z_k,\ \forall\, k \in \mathcal{K}^2 \tag{2.15}$$

$$x^e_{ijk} \in \{0,1\},\ \forall\, (i,j) \in \mathcal{A}^e, k \in \mathcal{K}^e, e \in \mathcal{E} \tag{2.16}$$

$$y_i \in \{0,1\},\ \forall\, i \in \mathcal{N}^1 \tag{2.17}$$

$$w^2_{si} \in \{0,1\},\ \forall\, s \in \mathcal{S}, i \in \mathcal{C} \tag{2.18}$$

$$z_k \in \{0,1\},\ \forall\, k \in \mathcal{K}^1 \cup \mathcal{K}^2 \tag{2.19}$$

$$g_{psk} \geq 0,\ \forall\, p \in \mathcal{P}, s \in \mathcal{S}, k \in \mathcal{K}^1 \tag{2.20}$$

$$u^e_i \in [1, |\mathcal{D}^e|],\ \forall\, i \in \mathcal{N}^e, e \in \mathcal{E}. \tag{2.21}$$

The objective function (2.1) aims to minimize facilities and vehicles fixed costs as well as distance-related costs. Constraints (2.2) are vehicle flow conservation constraints for both echelons. Constraints (2.3) are Miller-Tucker-Zemlin (MTZ) subtour elimination constraints for both echelons (Miller; Tucker; Zemlin, 1960). Constraints (2.4) ensure that each vehicle performs a single route. Constraints (2.5) define that a satellite is opened if and only if an FE vehicle leaves it. Constraints (2.6) state that every customer is visited exactly once. Constraints (2.7) define that each customer is assigned to exactly one satellite. Constraints (2.8) ensure that if a customer is assigned to a satellite, the vehicle that serves it leaves the corresponding satellite. Constraints (2.9) define that the amount of load transferred from a platform to a satellite is equal to the demand of the customers assigned to this satellite. Constraints (2.10) and (2.11) ensure that the capacities of the platforms and satellites are respected. Constraints (2.12) and (2.13) define that there is a load flow from a platform to a satellite only if they are both served by the same vehicle. Constraints (2.14) and (2.15) make sure that the vehicles' capacities are respected. Constraints (2.16)–(2.21) define the variables' domains. This formulation has $O((|\mathcal{P}| + |\mathcal{S}|)^2|\mathcal{K}^1| + (|\mathcal{S}| + |\mathcal{C}|)^2|\mathcal{K}^2|)$ variables and $O((|\mathcal{P}| + |\mathcal{S}|)^2 + (|\mathcal{S}| + |\mathcal{C}|)^2 + (|\mathcal{P}| + |\mathcal{S}|)|\mathcal{K}^1| + (|\mathcal{S}| + |\mathcal{C}|)|\mathcal{K}^2| + |\mathcal{P}||\mathcal{S}||\mathcal{K}^1| + |\mathcal{S}||\mathcal{C}||\mathcal{K}^2|)$ constraints.

It is worth noting that the original formulation has two minor issues that are corrected in CF1. First, Boccia et al. (2011) do not consider the echelon related index of variable u^e_i. Hence, for the satellites, these variables become poorly defined, since they appear in the constraints of both echelons. Additionally, in their paper, constraints (2.13) use Q^2 instead of Q^1, which is incorrect since they are related to the FE.

Improved formulation

A possible improvement to this formulation is the substitution of constraints (2.3), (2.7)–(2.15), and (2.18)–(2.21) by a commodity flow based formulation (Gavish; Graves, 1978). To this extent, we define continuous variables g^e_{ij} that represent the flow of commodities in arc $(i,j) \in \mathcal{A}^e, e \in \mathcal{E}$. The new formulation (ICF1) becomes:

$$\text{(ICF1) min} \sum_{i \in \mathcal{N}^1} H_i y_i + \sum_{e \in \mathcal{E}} \sum_{k \in \mathcal{K}^e} \sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} f^e x^e_{ijk} + \sum_{e \in \mathcal{E}} \sum_{k \in \mathcal{K}^e} \sum_{(i,j) \in \mathcal{A}^e} c^e_{ij} x^e_{ijk} \tag{2.22}$$

s.t. (2.2), (2.4)–(2.6), (2.16)–(2.17)

$$\sum_{i:(i,s) \in \mathcal{A}^1} g^1_{is} - \sum_{i:(s,i) \in \mathcal{A}^1} g^1_{si} = -\sum_{j \in \mathcal{C}} g^2_{js}, \ \forall\, s \in \mathcal{S} \tag{2.23}$$

$$\sum_{i:(i,j) \in \mathcal{A}^2} g^2_{ij} - \sum_{i:(j,i) \in \mathcal{A}^2} g^2_{ji} = -q_j, \ \forall\, j \in \mathcal{C} \tag{2.24}$$

$$\sum_{i \in \mathcal{D}^e} g^e_{ij} \leq B_j y_j, \ \forall\, j \in \mathcal{O}^e, e \in \mathcal{E} \tag{2.25}$$

$$0 \leq g^e_{ij} \leq Q^e \sum_{k \in \mathcal{K}^e} x^e_{ijk}, \ \forall\, (i,j) \in \mathcal{A}^e, e \in \mathcal{E}. \tag{2.26}$$

The objective function (2.22) is equivalent to (2.1) but uses a different form for calculating vehicle fixed costs. Constraints (2.23) define that the difference between the load arriving and leaving a satellite is the load transshipped through this satellite. Constraints (2.24) do the same for the customers. Constraints (2.25) ensure that the facilities capacities are respected. Constraints (2.26) define the domain of the new decision variables. The number of variables in ICF1 is of the same order as in CF1, but the number of constraints is significantly reduced to $O((|\mathcal{P}| + |\mathcal{S}|)|\mathcal{K}^1| + (|\mathcal{S}| + |\mathcal{C}|)|\mathcal{K}^2|)$.

Valid inequalities

It is well-known that vehicle index formulations for routing problems exhibit solution symmetries, which may negatively impact the performance of branch-and-bound-based methods (Furtado; Munari; Morabito, 2017; Munari; Savelsbergh, 2022). To mitigate this issue, one could add the following valid inequalities (VIs):

$$\sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} x^e_{ijk} \geq \sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} x^e_{ij(k+1)}, \ \forall\, k \in \mathcal{K}^e \setminus \{|\mathcal{K}^e|\}, e \in \mathcal{E} \tag{2.27}$$

$$\sum_{(i,j) \in \mathcal{A}^2 : i<h} x^2_{ij(k-1)} \geq \sum_{j:(h,j) \in \mathcal{A}^2} x^2_{hjk}, \ \forall\, h \in \mathcal{C} \setminus \{1\}, k \in \mathcal{K}^2 \setminus \{1\}. \tag{2.28}$$

Constraints (2.27) state that, if a vehicle is used, a vehicle with a smaller index is also used. Constraints (2.28) ensure that, if a vehicle serves a customer, a vehicle with a smaller index serves another customer with a smaller index.

In addition to these, the following VIs could be used to tighten the LR of the formulations:

$$\sum_{i \in \mathcal{O}^e} y_i \geq o^e_{min}, \ \forall \, e \in \mathcal{E} \tag{2.29}$$

$$\sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} \sum_{k \in \mathcal{K}^e} x^e_{ijk} \geq \left\lceil \frac{1}{Q^e} \sum_{i \in \mathcal{C}} q_i \right\rceil, \ \forall \, e \in \mathcal{E} \tag{2.30}$$

$$\sum_{k \in \mathcal{K}^e} \sum_{j:(i,j) \in \mathcal{A}^e} x^e_{ijk} \geq y_i, \ \forall \, i \in \mathcal{O}^e, e \in \mathcal{E} \tag{2.31}$$

$$2 \sum_{k \in \mathcal{K}^1} x^1_{psk} \leq y_p + y_s, \ \forall \, p \in \mathcal{P}, s \in \mathcal{S} \tag{2.32}$$

$$2 \sum_{k \in \mathcal{K}^1} x^1_{spk} \leq y_p + y_s, \ \forall \, p \in \mathcal{P}, s \in \mathcal{S} \tag{2.33}$$

$$\sum_{k \in \mathcal{K}^2} x^2_{sjk} \leq y_s, \ \forall \, s \in \mathcal{S}, j \in \mathcal{C} \tag{2.34}$$

$$\sum_{k \in \mathcal{K}^2} x^2_{jsk} \leq y_s, \ \forall \, s \in \mathcal{S}, j \in \mathcal{C} \tag{2.35}$$

$$\sum_{i \in \mathcal{C}} w^2_{si} \geq y_s, \ \forall \, s \in \mathcal{S} \tag{2.36}$$

$$w^2_{si} \leq y_s, \ \forall \, s \in \mathcal{S}, i \in \mathcal{C}. \tag{2.37}$$

Constraints (2.29) define lower bounds on the number of platforms and satellites (Yıldız; Karaoğlan; Altiparmak, 2023). In these VIs, o^e_{min} are lower bounds on the number of facilities opened and can be defined by ordering the corresponding facilities in decreasing order of capacity and taking the smallest number of them that can serve all the customers' demands. Constraints (2.30) are lower bounds on the number of vehicles needed in each echelon. Constraints (2.31) define that if a facility is opened, at least one vehicle leaves it. Constraints (2.32) and (2.33) forbid vehicles from traveling between a platform and a satellite if one of them is not opened. Constraints (2.34) and (2.35) state that a vehicle can only leave from or return to a satellite if it is opened. Constraints (2.36) state that, if a satellite is opened, at least one customer is assigned to it. Constraints (2.37) forbid customers to be assigned to satellites that are not opened. It is worth noticing that VIs (2.36) and (2.37) cannot be used with ICF1 because variables w^2 are not defined in this formulation.

2.3.2 Formulation with two-index arc variables and binary assignments (CF2)

We introduce a novel formulation for the 2E-LRP with two-index routing variables. This formulation is based on a 2E-VRP formulation (Belgin; Karaoğlan; Altiparmak, 2018) that has been adapted to the 2E-LRP by Yıldız, Karaoğlan, and Altiparmak (2023). However, when adapting it to the 2E-LRP, they did not include the platforms' capacity constraints since it would

create nonlinearities. We have adapted it by considering commodity flow variables in both echelons to ensure that these capacities are respected.

In this formulation, the binary variable x_{ij}^e indicates whether a vehicle traverses arc $(i,j) \in \mathcal{A}^e$ in echelon $e \in \mathcal{E}$. As mentioned in Section 2.1, when avoiding the variables with vehicle index, additional variables and constraints must be used to ensure that the vehicles end their routes in the facilities they started from. In this formulation, this is made by the variable w_{si}^2 already employed in CF1 and the binary variable w_{ps}^1 that indicates whether satellite $s \in \mathcal{S}$ is assigned to platform $p \in \mathcal{P}$.

The first formulation with two-index arc variables is defined as:

$$\text{(CF2) min} \sum_{i \in \mathcal{N}^1} H_i y_i + \sum_{e \in \mathcal{E}} \sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} f^e x_{ij}^e + \sum_{e \in \mathcal{E}} \sum_{(i,j) \in \mathcal{A}^e} c_{ij}^e x_{ij}^e \tag{2.38}$$

$$\text{s.t. (2.7), (2.17), (2.23)–(2.25)}$$

$$\sum_{i:(i,j) \in \mathcal{A}^e} x_{ij}^e = \sum_{i:(j,i) \in \mathcal{A}^e} x_{ji}^e, \ \forall\, j \in \mathcal{N}^e, e \in \mathcal{E} \tag{2.39}$$

$$\sum_{j:(s,j) \in \mathcal{A}^1} x_{sj}^1 = y_s, \ \forall\, s \in \mathcal{S} \tag{2.40}$$

$$\sum_{j:(i,j) \in \mathcal{A}^2} x_{ij}^2 = 1, \ \forall\, i \in C \tag{2.41}$$

$$\sum_{p \in P} w_{ps}^1 = y_s, \ \forall\, s \in \mathcal{S} \tag{2.42}$$

$$x_{ij}^e \leq w_{ij}^e, \ \forall\, i \in \mathcal{O}^e, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.43}$$

$$x_{ji}^e \leq w_{ij}^e, \ \forall\, i \in \mathcal{O}^e, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.44}$$

$$x_{ij}^e + w_{hi}^e + \sum_{h' \in \mathcal{O}^e \setminus \{h\}} w_{h'j}^e \leq 2, \ \forall\, i,j \in \mathcal{D}^e, i \neq j, h \in \mathcal{O}^e \tag{2.45}$$

$$x_{ij}^e \in \{0,1\}, \ \forall\, (i,j) \in \mathcal{A}^e, e \in \mathcal{E} \tag{2.46}$$

$$w_{ij}^e \in \{0,1\}, \forall\, i \in \mathcal{O}^e, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.47}$$

$$0 \leq g_{ij}^e \leq Q^e x_{ij}^e, \ \forall\, (i,j) \in \mathcal{A}^e, e \in \mathcal{E}. \tag{2.48}$$

The objective function (2.38) and constraints (2.39)–(2.41) are the two-index variables equivalent of (2.22), (2.2), (2.5), and (2.6), respectively. Constraints (2.42) define that if a satellite is opened, it is assigned to a platform. Constraints (2.43) and (2.44) state that if a vehicle travels between an origin and a destination, this destination is assigned to this origin. Constraints (2.45) ensure that a vehicle can only travel between two destinations assigned to the same origin. Constrains (2.46)–(2.48) define the domain of variables. This formulation has $O((|\mathcal{P}| + |\mathcal{S}|)^2 + (|\mathcal{S}| + |\mathcal{C}|)^2)$ variables and $O(|\mathcal{P}||\mathcal{S}|^2 + |\mathcal{S}||\mathcal{C}|^2 + (|\mathcal{P}| + |\mathcal{S}|)^2 + (|\mathcal{S}| + |\mathcal{C}|)^2)$ constraints.

Improved formulation

The first possible improvement to CF2 concerns constraints (2.45). They were presented this way since it is the common approach in the literature (Belgin; Karaoğlan; Altiparmak, 2018; Yıldız; Karaoğlan; Altiparmak, 2023). However, they can be improved to become sparser and provide a tighter LR. For the SE, from constraints (2.7), we have that $\sum_{h' \in \mathcal{O}^2 \setminus \{h\}} w^2_{h'j} = 1 - w^2_{hj}$ and this can be substituted in constraints (2.45) to make them sparser. For the FE, we would have $\sum_{h' \in \mathcal{O}^1 \setminus \{h\}} w^1_{h'j} = y_h - w^1_{hj}$ from constraints (2.42), but it is possible to use $1 - w^1_{hj}$ because constraints (2.45) are redundant for $y_h = 0$. Moreover, given that constraints (2.45) define that a vehicle may travel between two destinations only if they are both assigned to the same origin, we can add x^e_{ji} to their left-hand side, tightening the LR. This way, we obtain the following formulation ICF2:

(ICF2) min (2.38)

s.t. (2.7), (2.17), (2.23)–(2.25), (2.39)–(2.44), (2.46)–(2.48)

$$x^e_{ij} + x^e_{ji} + w^e_{hi} - w^e_{hj} \leq 1, \ \forall \, i, j \in \mathcal{D}^e, i \neq j, h \in \mathcal{O}^e. \tag{2.49}$$

Valid inequalities

Both CF2 and ICF2 can be enhanced by the following VIs:

(2.29), (2.36)–(2.37)

$$\sum_{i \in \mathcal{O}^e} \sum_{j \in \mathcal{D}^e} x^e_{ij} \geq \left\lceil \frac{1}{Q^e} \sum_{c \in \mathcal{C}} q_c \right\rceil, \ \forall \, e \in \mathcal{E} \tag{2.50}$$

$$\sum_{j \in \mathcal{D}^e} x^e_{ij} \geq y_i, \ \forall \, i \in \mathcal{O}^e, e \in \mathcal{E} \tag{2.51}$$

$$2x^1_{ps} \leq y_p + y_s, \ \forall \, p \in \mathcal{P}, s \in \mathcal{S} \tag{2.52}$$

$$2x^1_{sp} \leq y_p + y_s, \ \forall \, p \in \mathcal{P}, s \in \mathcal{S} \tag{2.53}$$

$$x^2_{sj} \leq y_s, \ \forall \, s \in \mathcal{S}, j \in \mathcal{C} \tag{2.54}$$

$$x^2_{js} \leq y_s, \ \forall \, s \in \mathcal{S}, j \in \mathcal{C} \tag{2.55}$$

$$\sum_{s \in \mathcal{S}} w^1_{ps} \geq y_p, \ \forall \, p \in \mathcal{P} \tag{2.56}$$

$$2w^1_{ps} \leq y_p + y_s, \ \forall \, p \in \mathcal{P}, s \in \mathcal{S}. \tag{2.57}$$

Constraints (2.50)–(2.55) are the two-index variables equivalent to (2.30)–(2.35). Constraints (2.56) define that if a platform is opened at least one satellite is assigned to it. Constraints (2.57) define that a satellite can only be assigned to a platform if both the satellite and the platform are opened. Constraints (2.50) and (2.52)–(2.56) can be found in Yıldız, Karaoğlan, and Altiparmak (2023).

2.3.3 Formulation with two-index arc variables and continuous assignments (CF3)

In this section, we present a novel two-index formulation that does not require the assignment variables w from CF2. Instead of binary assignments, this formulation is based on continuous variables v_j^e that indicate from which origin the vehicle visiting destination $j \in \mathcal{D}^e, e \in \mathcal{E}$ departed. It is inspired by the index propagation formulation of Furtado, Munari, and Morabito (2017) for the pickup and delivery routing problem. Formulation CF3 is defined as:

(CF3) min (2.38)

s.t. (2.17), (2.23)–(2.25), (2.39)–(2.41), (2.46), (2.48)

$$v_j^e \geq \sum_{i \in \mathcal{O}^e} i x_{ij}^e, \ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.58}$$

$$v_j^e \geq \sum_{i \in \mathcal{O}^e} i x_{ji}^e, \ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.59}$$

$$v_j^e \leq M_1^e - \sum_{i \in \mathcal{O}^e} (M_1^e - i) x_{ij}^e, \ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.60}$$

$$v_j^e \leq M_1^e - \sum_{i \in \mathcal{O}^e} (M_1^e - i) x_{ji}^e, \ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.61}$$

$$v_j^e \geq v_i^e - M_2^e (1 - x_{ij}^e - x_{ji}^e), \ \forall\, i, j \in \mathcal{D}^e, i \neq j, e \in \mathcal{E} \tag{2.62}$$

$$1 \leq v_j^e \leq |\mathcal{O}^e|, \ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E}. \tag{2.63}$$

Constraints (2.58)–(2.61) impose that if a vehicle travels between an origin i and a destination j, then v_j^e assumes the value of i, this way indicating the origin related to node j. Constraints (2.62) ensure that if a vehicle travels between two destinations i and j, then these nodes are in the same route and, therefore, have the same origin (i.e., $v_i^e = v_j^e$). Constraints (2.63) define the domain of the new variables. M_1^e and M_2^e are sufficiently large numbers. Their tightest possible values are $M_1^e = |\mathcal{O}^e|$ and $M_2^e = |\mathcal{O}^e| - 1$. This formulation has $O((|\mathcal{P}| + |\mathcal{S}|)^2 + (|\mathcal{S}| + |\mathcal{C}|)^2)$ variables and $O((|\mathcal{P}| + |\mathcal{S}|)^2 + (|\mathcal{S}| + |\mathcal{C}|)^2)$ constraints.

Formulation CF3 can be enhanced by the following VIs:

(2.29), (2.50)–(2.55)

$$v_j^e \geq i \left(y_i - \sum_{i' \in \mathcal{O}^e : i' < i} y_{i'} \right), \ \forall\, i \in \mathcal{O}^e \setminus \{1\}, j \in \mathcal{D}^e, e \in \mathcal{E} \tag{2.64}$$

$$v_j^e \leq i + \sum_{i' \in \mathcal{O}^e : i' > i} i' y_{i'} + (|\mathcal{O}^e| - i)(1 - y_i), \ \forall\, i \in \mathcal{O}^e \setminus \{|\mathcal{O}^e|\}, j \in \mathcal{D}^e, e \in \mathcal{E}. \tag{2.65}$$

Constraints (2.64) define that destinations are assigned to an origin with an index at least equal to the smallest index of opened origins. Analogously, constraints (2.65) ensure that destinations are assigned to an origin with an index at most equal to the greatest index of opened

origins.

2.4 Comparison of LRs

In this section, we discuss some relationships between the LRs of the different proposed formulations. Propositions 2.1 to 2.6 and Corollaries 2.1 to 2.4 enunciate and prove them.

Proposition 2.1. *The LR of ICF1 is not weaker than that of CF1.*

Proof. The optimal value of the LR of ICF1 for instance "100–10MN" from set *Nguyen*[2] is 156,294, higher than that of CF1, which is 111,867. □

Proposition 2.2. *Formulation ICF2 has a stronger LR than formulation CF2.*

Proof. The optimal value of the LR of ICF2 for instance "100–10MN" from set *Nguyen* is 160,148, which is higher than that of the LR of CF2 for the same instance (156,294). Hence, CF2 does not have a stronger LR than ICF2.

The fact that the LR of ICF2 is stronger than that of CF2 comes directly from the fact that, if constraints (2.49) are satisfied, constraints (2.45) are also satisfied. Indeed, from constraints (2.7) and (2.42), $\sum_{h' \in \mathcal{O}^e \setminus \{h\}} w^e_{h'j} - 1 = -w^e_{hj}$. Substituting this in (2.49) makes

$$1 \geq x^e_{ij} + x^e_{ji} + w^e_{hi} - w^e_{hj} \geq x^e_{ij} + w^e_{hi} + \sum_{h' \in \mathcal{O}^e \setminus \{h\}} w^e_{h'j} - 1, \ \forall\, i, j \in \mathcal{D}^e, i \neq j, h \in \mathcal{O}^e,$$

corresponding precisely to constraints (2.45). □

Proposition 2.3. *Formulation CF2 has a stronger LR than formulation ICF1.*

Proof. Given a solution $\overline{x}^e_{ij}, (i,j) \in \mathcal{A}^e, e \in \mathcal{E}$, for the LR of CF2, it is possible to define a solution for the LR of ICF1 by making $\widetilde{x}^e_{ijk} = \frac{1}{|\mathcal{K}^e|}\overline{x}^e_{ij}, (i,j) \in \mathcal{A}^e, k \in \mathcal{K}^e, e \in \mathcal{E}$. This way, we have that constraints (2.39) ⇒ (2.2), (2.40) ⇒ (2.5), (2.41) ⇒ (2.6), (2.46) ⇒ (2.16), and (2.48) ⇒ (2.26). Moreover, (2.40) and (2.41) ⇒ (2.4). In fact, from (2.41),

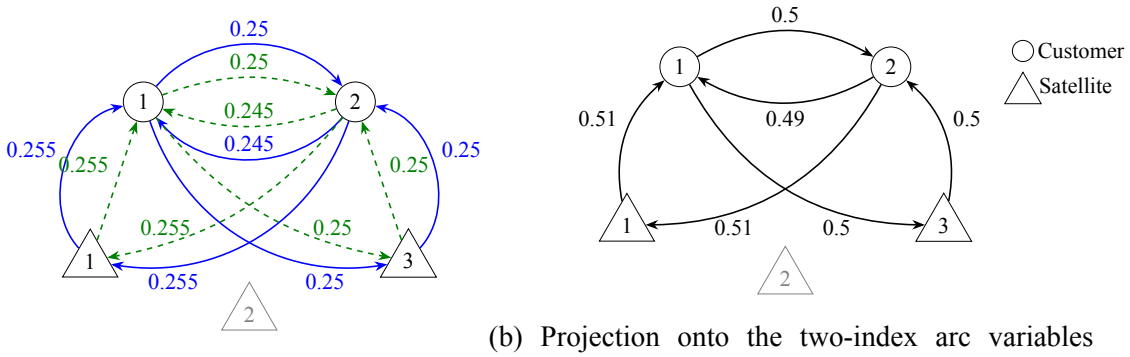
$$\sum_{j \in \mathcal{C} \setminus \{i\}} \overline{x}^2_{ij} + \sum_{j \in \mathcal{S}} \overline{x}^2_{ij} = 1, \ \forall\, i \in \mathcal{C} \Rightarrow \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{S}} \overline{x}^2_{ij} = |\mathcal{C}| - \sum_{i,j \in \mathcal{C}: i \neq j} \overline{x}^2_{ij} \Rightarrow$$

$$\Rightarrow \sum_{i \in \mathcal{C}} \sum_{j \in \mathcal{S}} \widetilde{x}^2_{ijk} \leq \frac{|\mathcal{C}|}{|\mathcal{K}^2|} \leq 1, \ \forall\, k \in \mathcal{K}^2$$

since the fleet is unlimited, corresponding precisely to constraints (2.4) for the SE. For the FE, the derivation from (2.40) is analogous. Hence, it is proved that, if $\overline{x}$ is a solution to the LR of CF2, then $\widetilde{x}$ is a solution to the LR of ICF1.

[2]The benchmark instances sets are properly presented in Section 2.5.

Figure 2.2 represents an example of a solution of the LR of ICF1 that is not a solution of the LR of CF2. It presents an instance with three potential satellites and two customers (the FE is not presented since it is not needed in this demonstration). The customers in this instance have low demands, while vehicles and facilities have large enough capacities to ensure that load and capacity constraints are non-binding. Figure 2.2a portrays a solution of the LR of ICF1. The solid blue arrows represent the positive arc variables associated with one vehicle and the dashed green ones represent another vehicle. It is easy to see that these values respect all constraints of ICF1.



(a) Solution of the LR of ICF1.

(b) Projection onto the two-index arc variables space.

Figure 2.2: An example for proving that ICF1 does not have a stronger LR than CF2.

The only way to project this solution onto the two-index arc variables space of CF2 while respecting constraints (2.41) is by defining $x^2_{ij} = \sum_{k \in \mathcal{K}^2} \overline{x}^2_{ijk}, \ \forall \ (i,j) \in \mathcal{A}^2$, yielding the solution shown in Figure 2.2b. However, constraints (2.43) and (2.44) would impose $w^2_{1,1} \geq 0.51$ and $w^2_{3,1} \geq 0.5$. This implies $\sum_{s \in \mathcal{S}} w^2_{s1} \geq 1.01$, violating constraints (2.7). Hence, this solution of the LR of ICF1 does not have a correspondent solution for the LR of CF2.

□

Corollary 2.1. *Formulation ICF2 has a stronger LR than formulation ICF1.*

Corollary 2.2. *The LR of CF2 is not weaker than the LR of CF1.*

Corollary 2.3. *The LR of ICF2 is not weaker than the LR of CF1.*

Proposition 2.4. *Formulation CF3 has a stronger LR than formulation ICF1.*

Proof. The proof that every feasible solution for the LR of CF3 has a corresponding feasible solution in the LR of ICF1 is the same as in the proof of Proposition 2.3 for CF2 and ICF1. Also, the optimal value of the LR of ICF1 for instance "100–10MN" from set *Nguyen* is 156,294, while for the LR of CF3 it is 160,146. □

Corollary 2.4. *The LR of CF3 is not weaker than the LR of CF1.*

Proposition 2.5. *The LRs of formulations CF2 and CF3 are not comparable.*

Proof. The optimal value of the LR of CF3 for instance "100–10MN" from set *Nguyen* is 160,146 while for the LR of CF2 it is 156,294. Hence, the LR of CF2 is not stronger than that of CF3.

Moreover, Figure 2.2b presents an example of a solution of the LR of CF3 that is not a solution for the LR of CF2. Once again, assume that the customers have low demands and the vehicles and facilities have high enough capacities. The arrows represent the value of the corresponding x^2 variables. It is easy to see that $v_1^2 = v_2^2 = 1.5$ is a solution to CF3. However, for CF2, constraints (2.43) and (2.44) would impose $w_{1,1}^2 \geq 0.51$ and $w_{3,1}^2 \geq 0.5$. This implies $\sum_{s\in\mathcal{S}} w_{s1}^2 \geq 1.01$, violating constraints (2.7). Hence, this solution of the LR of CF3 is not feasible for the LR of CF2. □

Proposition 2.6. *Formulation ICF2 has a stronger LR than formulation CF3 if* $M_1^e = M_2^e = \frac{|\mathcal{O}^e|(|\mathcal{O}^e|+1)}{2}$.

Proof. First, we prove that a solution in the LR of ICF2 has a corresponding solution in the LR of CF3 by defining $v_j^e = \sum_{i\in\mathcal{O}^e} i w_{ij}^e$, which automatically respects constraints (2.63). By multiplying constraints (2.43) and (2.44) by i and summing over $\mathcal{O}^e$, we get

$$\sum_{i\in\mathcal{O}^e} i w_{ij}^e \geq \sum_{i\in\mathcal{O}^e} i x_{ij}^e,\ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E},\ \text{and}$$
$$\sum_{i\in\mathcal{O}^e} i w_{ij}^e \geq \sum_{i\in\mathcal{O}^e} i x_{ji}^e,\ \forall\, j \in \mathcal{D}^e, e \in \mathcal{E},$$

which correspond to constraints (2.58) and (2.59), respectively.

From constraints (2.41), in the SE, we have

$$\sum_{s'\in\mathcal{S}} x_{is'}^2 + \sum_{j\in\mathcal{C}\setminus\{i\}} x_{ij}^2 = 1,\ \forall\, i \in \mathcal{C} \Rightarrow x_{is}^2 + \sum_{j\in\mathcal{C}\setminus\{i\}} x_{ij}^2 = 1 - \sum_{s'\in\mathcal{S}\setminus\{s\}} x_{is'}^2,\ \forall\, s \in \mathcal{S}, i \in \mathcal{C}.$$

From constraints (2.7) and (2.43),

$$x_{is}^2 + \sum_{j\in\mathcal{C}\setminus\{i\}} x_{ij}^2 \geq 1 - \sum_{s'\in\mathcal{S}\setminus\{s\}} w_{s'i}^2 = w_{si}^2,\ \forall\, s \in \mathcal{S}, i \in \mathcal{C}.$$

By multiplying these inequalities by s and summing over $\mathcal{S}$, we get

$$\sum_{s\in\mathcal{S}} s w_{si}^2 \leq \sum_{s\in\mathcal{S}} s x_{is}^2 + \sum_{s\in\mathcal{S}} s \left(\sum_{j\in\mathcal{C}\setminus\{i\}} x_{ij}^2 \right),\ \forall\, i \in \mathcal{C}$$
$$\Rightarrow v_i^2 \leq \sum_{s\in\mathcal{S}} s x_{is}^2 + \frac{|\mathcal{S}|(|\mathcal{S}|+1)}{2} \left(1 - \sum_{s\in\mathcal{S}} x_{is}^2 \right),\ \forall\, i \in \mathcal{C},$$

which yields constraints (2.61) for $e = 2$. The derivations for constraints (2.60) and the FE are analogous.

Finally, by multiplying constraints (2.49) by h and summing over $\mathcal{O}^e$, we get

$$\frac{|\mathcal{O}^e|(|\mathcal{O}^e|+1)}{2}(x^e_{ij}+x^e_{ji})+\sum_{h\in\mathcal{O}^e} h(w^e_{hi}-w^e_{hj}) \leq \frac{|\mathcal{O}^e|(|\mathcal{O}^e|+1)}{2},\ \forall\, i,j\in\mathcal{D}^e, i\neq j,$$

which are precisely constraints (2.62).

The example presented in Figure 2.2b along with the discussion in the proof of Proposition 2.5 also works in this proposition to show that the LR of CF3 is not stronger than the LR of ICF2. □

Figure 2.3 illustrates the properties presented in Propositions 2.1 to 2.6 and in Corollaries 2.1 to 2.4. This figure does not explicitly represent all of these relationships since the strength of the LR is a transitive property, i.e., if formulation A has a stronger LR than another formulation B and the LR of B is stronger than that of C, the LR of A dominates that of C.

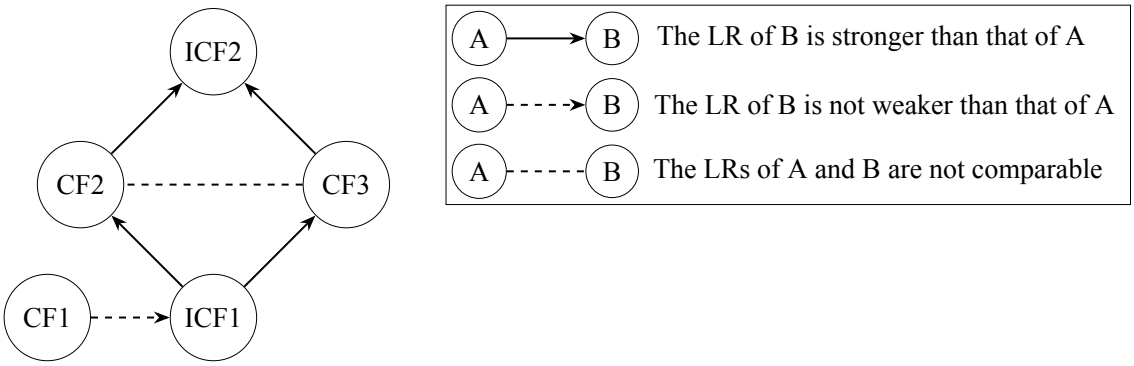


Figure 2.3: A visual representation of the relationships between different 2E-LRP formulations.

2.5 Computational experiments

This section presents the results of the extensive computational experiments developed to assess the performance of the presented formulations of the 2E-LRP in a general-purpose MIP solver. All experiments were run on a computing cluster from Compute Canada, where each node is equipped with 2xAMD Rome 7532 processors running at 2.4GHz. The formulations were implemented in C++ using Gurobi 11.0 as solver with an optimality tolerance of 10^{-7}. All experiments were limited to one hour of runtime and 80GB of RAM, using up to eight threads.

We performed experiments with five benchmark instance sets of the 2E-LRP and its variants. The first of them is the *Prodhon* set, which contains 30 instances with the number of customers ranging from 20 to 200, the number of potential satellites being five or 10, and the number of potential platforms fixed as one. The second instance set is called *Nguyen* and contains 24 instances with one platform in each, five or ten potential satellites, and a range of customers that goes from 25 to 200. These two instances sets were introduced by Nguyen, Prins, and Prodhon (2012b).

The remaining three instance sets, named *I1*, *I2*, and *I3*, were generated by Contardo, Hemmelmayr, and Crainic (2012) following the procedure suggested by Boccia et al. (2011). These sets contain 31 instances each with the number of potential platforms ranging from two to five, the number of potential satellites going from three to 20, and the number of customers between eight and 200.

We ran our experiments using all instances in these sets, except those with 200 customers. These instances were excluded because they are too large for CFs to handle, as few formulations found feasible solutions and only for few instances of this size. The remaining instances were divided into three groups: small (from eight to 25 customers), medium (from 50 to 75 customers), and large (from 100 to 150 customers).

In Section 2.5.1, the presented formulations are compared in terms of their LR, their performance, and their number of constraints and variables. They are also compared with the best known solutions (BKS) from the literature. Section 2.5.2 assesses how the existence or absence of multiple potential platforms in an instance affects the performance of the formulations. Finally, Section 2.5.3 evaluates the benefits of including valid inequalities for each formulation.

2.5.1 Comparison of CFs

The first assessment to be made is on how the different formulations compare to each other empirically. We also confront these results with the BKS from the literature, considering both the best known lower bounds (BKLBs) and the best known upper bounds (BKUBs). The BKLBs have all been presented by Contardo, Hemmelmayr, and Crainic (2012), while the BKUBs have been reported by Contardo, Hemmelmayr, and Crainic (2012), Nguyen, Prins, and Prodhon (2012b) and Schwengerer, Pirkwieser, and Raidl (2012), and Breunig et al. (2016). For each instance, we computed the gap of the BKS as $\frac{BKUB-BKLB}{BKUB}$. Instances with this gap equal to zero were considered having an optimal solution found. Although no BKUB was improved, the presented formulations found many of the reported BKUBs and improved most of the BKLBs.

Table 2.2 presents the results of different metrics for each formulation. These results are aggregated by size (small, medium, and large) and also by all instances. Detailed results are presented as supplementary material. In this table, “Size” indicates the instance size, “Metric” presents the corresponding value, “BKS” is the best known solution, and “CF1”, “ICF1”, “CF2”, “ICF2”, and “CF3” indicate the corresponding formulation. “LR” represents the optimal value of the LR of the corresponding model, “LB” and “UB” are respectively the lower and upper bounds reported by the solver at the end of the runtime, “Gap (%)” corresponds to the optimality gap, “Time (s)” indicates the runtime in seconds, and “# of optimals” indicates the number of instances with proved optimality. Except for “# of optimals”, all reported values are averages. Moreover, for the gap, the presented value is the average of optimality gaps, not the gap computed with the average LB and UB. For each instance, the gap reported for the BKS is defined as $\frac{BKUB-BKLB}{BKUB}$, while for the five formulations it is the optimality gap reported by

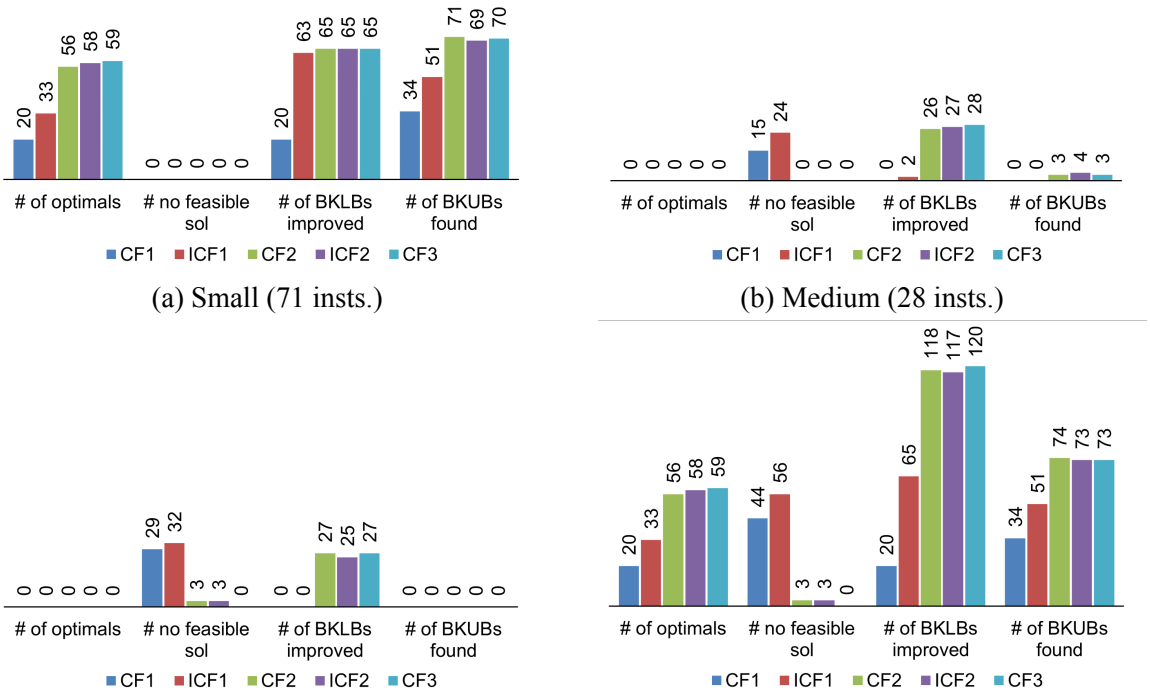
the solver $\left(\frac{UB-LB}{UB}\right)$. For the formulations that did not find feasible solutions to one or more instances of a given instance class, the corresponding UB and gap were reported as "N/A", since it is impossible to define these values for these specific instances.

Size	Metric	BKS	CF1	ICF1	CF2	ICF2	CF3
Small (71 insts.)	LR	–	5,139	6,715	6,715	6,924	6,921
	LB	8,506	7,880	8,741	8,930	8,930	8,931
	UB	8,934	9,090	8,947	8,934	8,934	8,934
	Gap (%)	7.78	14.11	3.02	0.38	0.36	0.33
	Time (s)	–	2,651	2,097	936	911	861
	# of optimals	6	20	33	56	58	59
Medium (28 insts.)	LR	–	35,855	50,174	50,174	51,308	51,303
	LB	61,850	50,018	59,569	64,659	64,860	64,910
	UB	67,071	N/A	N/A	68,022	67,743	67,641
	Gap (%)	9.88	N/A	N/A	8.46	6.75	6.37
	Time (s)	–	3,600	3,600	3,600	3,600	3,600
	# of optimals	0	0	0	0	0	0
Large (32 insts.)	LR	–	93,089	119,158	119,158	120,654	120,646
	LB	133,224	109,045	123,967	136,450	136,825	139,211
	UB	147,140	N/A	N/A	N/A	N/A	153,291
	Gap (%)	10.36	N/A	N/A	N/A	N/A	13.28
	Time (s)	–	3,602	3,600	3,600	3,600	3,600
	# of optimals	0	0	0	0	0	0
All (131 insts.)	LR	–	33,188	43,471	43,471	44,192	44,187
	LB	50,373	41,599	47,752	51,991	52,126	52,720
	UB	55,120	N/A	N/A	N/A	N/A	56,745
	Gap (%)	8.85	N/A	N/A	N/A	N/A	4.78
	Time (s)	–	3,086	2,786	2,156	2,142	2,116
	# of optimals	6	20	33	56	58	59

Table 2.2: Results of the MIP solver for different CFs.

Figure 2.4 delves deeper into the performances of the MIP solver for different CFs. In the four charts, "# of optimals" indicates the number of optimal solutions found, "# no feasible sol." corresponds to the number of instances for which the solver could not find a feasible solution, "# of BKLBs improved" and "# of BKUBs found" respectively indicate the number of instances to which the corresponding CF found an LB that was better than the BKLB or an UB that was as good as the BKUB. Figure 2.5 presents the average number of constraints, variables (any kind), and binary variables for each formulation ("# of constraints", "# of variables", and "# of binary variables", respectively). The results are aggregated by size and presented for the overall solution, as in Table 2.2.

Regarding the LR, it is clear that the CF1 presents the lowest values. On average, ICF1 has an LR bound that is 30.98% higher than that of CF1. Moreover, despite having a stronger LR than ICF1, CF2 presents the same result as ICF1 for all tested instances. The average value of the LR of CF3 is 1.65% higher than that of CF2, even though their LRs are not comparable. ICF2 has an average LR bound that is 1.66% higher than that of CF2, but only 0.01% higher than that of CF3. This indicates that, although ICF2 has a stronger LR than CF3, their difference may not



(a) Small (71 insts.)

(b) Medium (28 insts.)

(c) Large (32 insts.)

(d) All (131 insts.)

Figure 2.4: Results of the MIP solver for different CFs and sizes.

be significant in practice, since in the test instances they showed very similar results. Another impressive result is the fact that, for the medium and large instances, the average optimal values of LRs of ICF1, CF2, ICF2, and CF3 are higher than the average LB found by the solver after one hour of runtime with CF1 (this is also true for the overall average).

Comparing the number of variables and constraints, it is clear that ICF1 has a slightly larger number of general variables and smaller number of binary variables when compared to CF1. The number of constraints, however, is 81.21% smaller on average, significantly reducing the size of the linear programming problem solved in each branch-and-bound node. When comparing formulations CF2 and ICF2, the number of constraints increases again, being closer to that of CF1 and much larger than that of ICF1. The number of variables, however, is drastically reduced, going from being 88.95% smaller for the small-sized instances to being 98.22% smaller in the large-sized instances (for the binary variables the numbers are 93.57% and 99.07%, respectively). Finally, CF3 has the smallest number of variables and constraints of all formulations. Compared to ICF2, CF3 has 80.57% fewer constraints, 4.18% fewer variables, and 9.02% fewer binary variables, being the smallest formulation, while preserving almost all strength of the LR. This translates into the results of the MIP solver, since this is the CF with the best overall performance.

For the small instances, CF1 allows the solver to prove optimality for only 20 out of 71 of them, presenting an average gap of 14.11%. These results are considerably improved by ICF1,

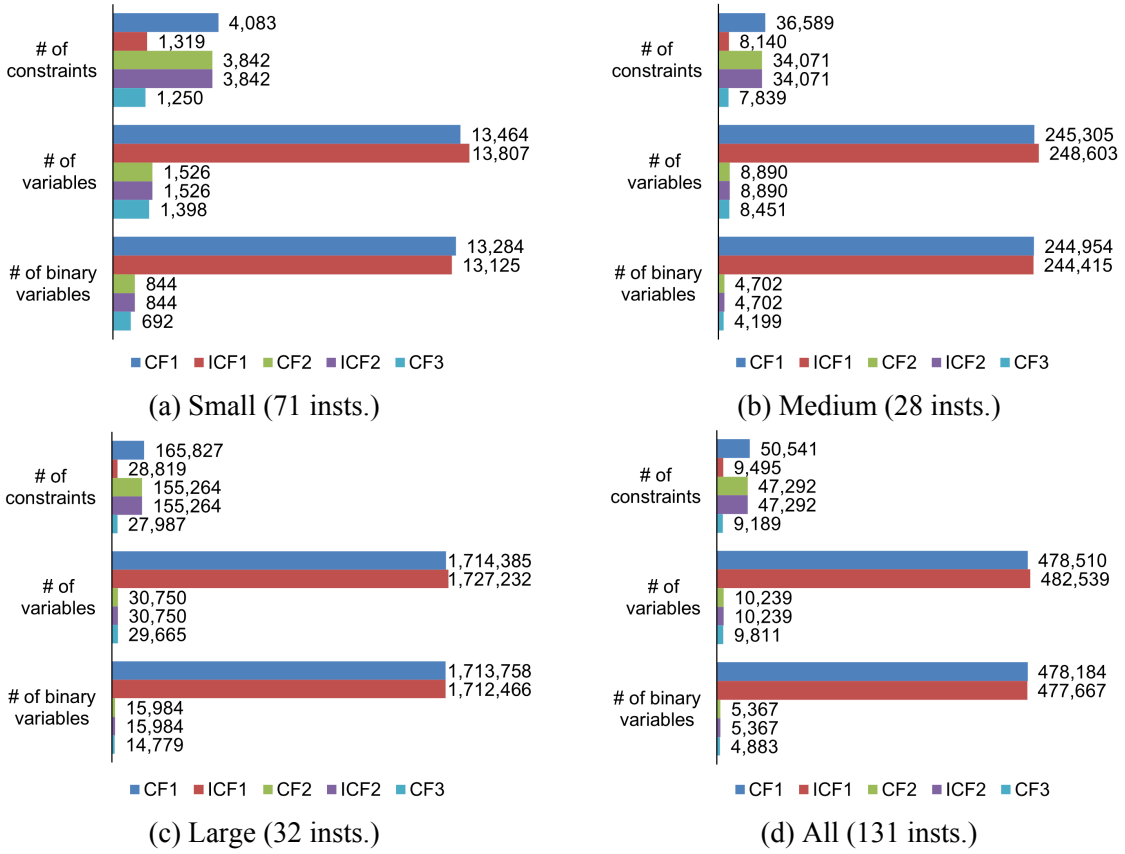


Figure 2.5: Average number of constraints, variables, and binary variables for different CFs and sizes.

as the solver proves optimality to a total of 33 instances, lowering the average gap to 3.02% and improving both the lower and the upper bounds. The three formulations with two-index arc variables (CF2, ICF2, and CF3) promote good results, with very similar UBs. The BKLB is improved in 65 instances, all of them with unknown optimal solutions in the literature. The solver finds the BKUB for all the 71 instances using CF2, for 69 instances using ICF2, and for 70 instances using CF3. The best LB is obtained when using CF3, which proves optimality for 59 of the instances (83.10% of them) and presents the best gap. The optimal solutions of 53 of these instances are not reported in the literature.

For medium and large instances, the solver can no longer find feasible solutions for all instances when using formulations with vehicle index variables. In fact, despite leading to better LBs on average, ICF1 results in feasible solutions for fewer instances than CF1. For medium-sized instances, the average gap for CF2 is 8.46%, whereas for ICF2 it is 1.71% lower, as a result of attaining better LBs and UBs. This fully justifies the modification in constraints (2.45) that yield ICF2 with sparser and stronger constraints. CF3 yields even better bounds and gap, with solutions that are only 0.85% worse than the BKUB on average, while improving the BKLB by 4.95%. CF3 also results in better BKLB for all of these instances, while finding the BKUB for three of them.

For the large instances, only CF3 leads to feasible solutions for all instances. The perfor-

mance is not as good as for the small- and medium-sized instances, since it presents 13.28% average gap. However, the obtained solutions are only 4.18% away from the average BKUB, which is an excellent result for a compact formulation in large-sized instances. Furthermore, the LB obtained with CF3 is 4.49% higher than the BKLB on average. In fact, the BKLB is improved for 27 out of 32 instances.

Overall, CF3 is the best performing CF and the only one that results in feasible solutions for all instances. The average gap is 4.78%, the UB is only 2.95% higher than the BKUB (which was obtained by tailored metaheurisitics) and the BKLB is improved in 4.66%, which is significant since it is a CF. It also leads to improved BKLB for 120 instances (91.60% of them), while resulting in the BKUB for 73 instances (55.73%).

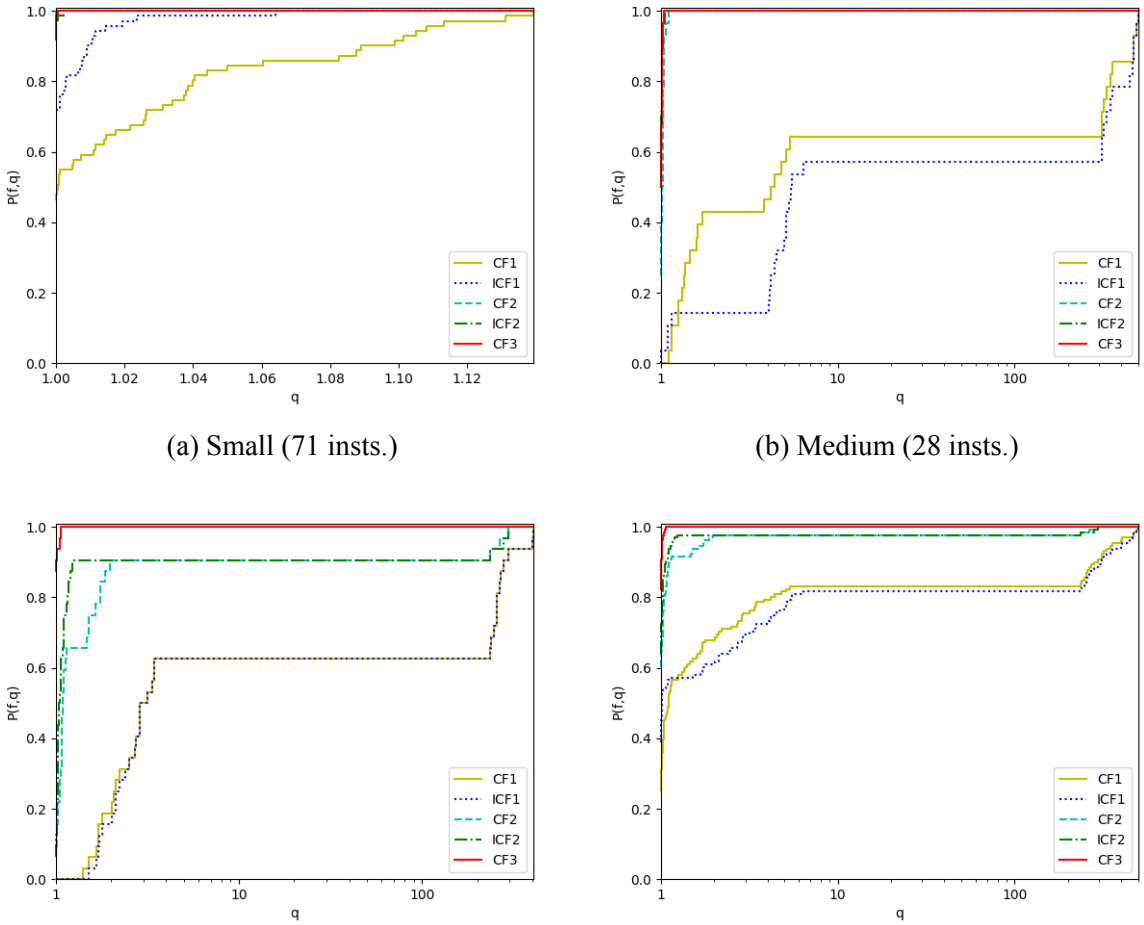
To further compare the performance of the MIP solver for different formulations, Figures 2.6 and 2.7 present performance profiles (Dolan; Moré, 2002) for the UB and optimality gap, respectively. For the UB, for example, given a set of instances and a set of CFs, denote by UB_{fp} the UB for instance p when solved with formulation f. In these graphs, for a value $q > 0$, $P(f, q)$ indicates the fraction of instances for which CF f finds solutions with an UB that lies within a factor q of the best obtained UB. Hence, the value of $P(f, 0)$ indicates the fraction of instances for which CF f finds the best UB among all CFs. For the gap, the definition is analogous. The graphs are presented with the horizontal axis in logarithmic scale.

For the UB, the performance profiles indicate that ICF1 outperforms CF1 for the small instances and is outperformed by CF1 for the medium ones, while for the large ones they are practically equivalent. On the overall average, CF1 slightly outperforms ICF1, which is coherent with the results in Figure 2.4, since there are more instances for which the solver does not find any feasible solution for ICF1 than for CF1. Nevertheless, for the gap, this behavior is not the same. Although for the small, medium, and large instances the comparison of CF1 and ICF1 is similar for both UB and gap, on the overall average, ICF1 outperforms CF1, since it provides much better LBs.

The performance profiles of CF2 and ICF2 for UB are very similar. They are practically the same for the small and medium instances, and, for the large instances, ICF2 outperforms CF2 for small values of q. For the gap, this difference is more significant. For the small instances, there is a small difference, which did not exist for the UB performance profiles. Moreover, for the medium and large instances, as well as for the overall average, this difference is more expressive due mostly to the improvements in the LR and the LB documented in Table 2.2.

Finally, the performance profiles confirm the results presented in Table 2.2 that CF3 is the best performing CF. For the UB, $P(f, q)$ is equal to one for almost every possible value of q, greatly outperforming the other CFs for the overall average. For the gap, the results of the MIP solver for CF3 are also much better than those for the other CFs.

Therefore, the presented results make clear that the use of formulations with vehicle index variables does not lead to good results in a general-purpose MIP solver. Moreover, in medium- and large-sized instances, it may not possible to obtain even feasible solutions. The two-index



(a) Small (71 insts.)

(b) Medium (28 insts.)

(c) Large (32 insts.)

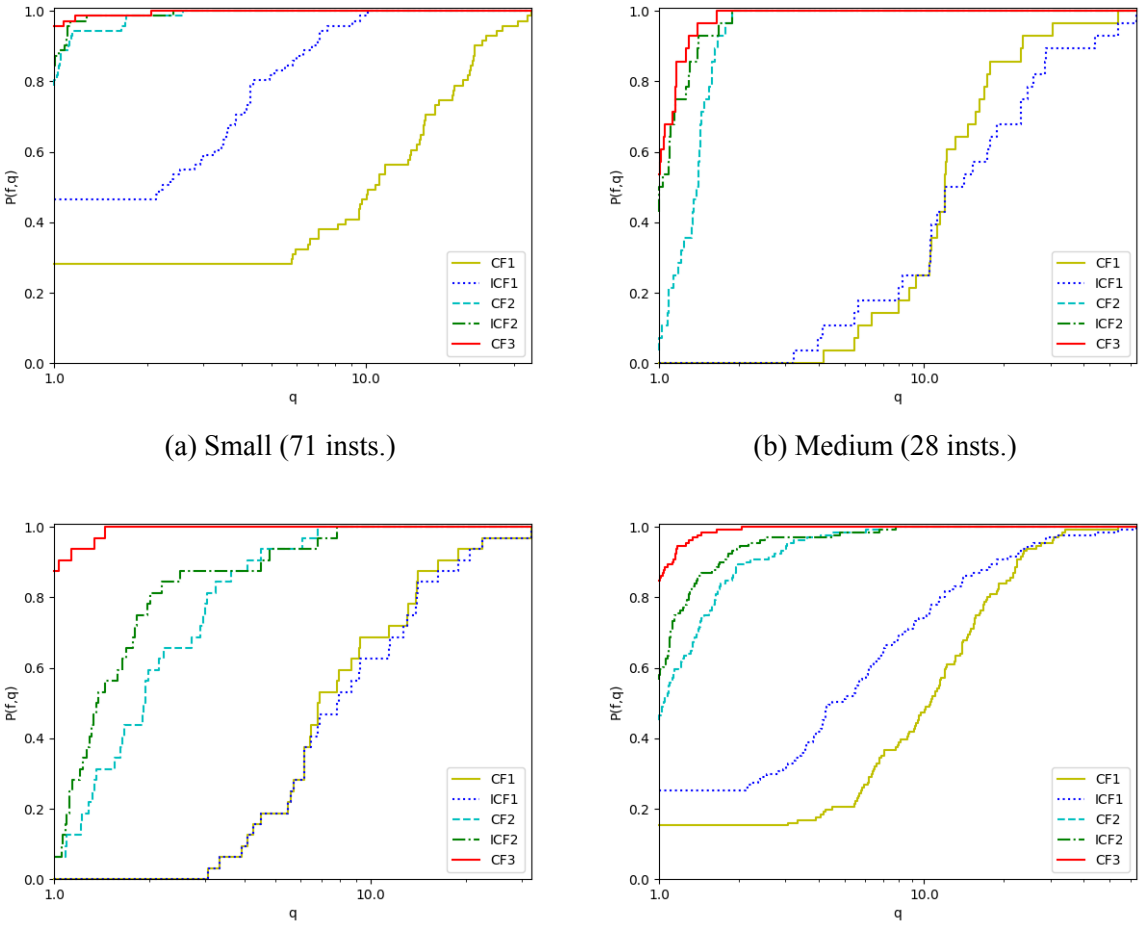
(d) All (131 insts.)

Figure 2.6: Performance profile of the MIP solver for different CFs and sizes considering the UB.

arc variables are more suited to solve the 2E-LRP, leading to much better results. Nevertheless, CF2, which is based in a formulation found in the literature, has too many constraints and binary variables, deteriorating the performance of the solver. CF3 is much smaller, while preserving most of the quality of the LR and being, therefore, the best option to represent the 2E-LRP with a compact formulation.

2.5.2 The impact of multiple platforms

Table 2.3 presents a closer look at how the number of platforms may affect the performance of the solver according to the addressed CFs. As discussed in Section 2.5, there are two instance sets with a single platform in each instance and three sets with multiple platforms. In Table 2.3, this information is presented in column $|\mathcal{P}|$. The results presented in the table clearly indicate that the number of platforms significantly affects the performance of the solver. In general, instances with a single platform are easier to solve than the ones with multiple platforms. This



(a) Small (71 insts.)

(b) Medium (28 insts.)

(c) Large (32 insts.)

(d) All (131 insts.)

Figure 2.7: Performance profile of the MIP solver for different CFs and sizes considering the optimality gap.

characteristic affects the solver's ability to find feasible solutions and prove optimality. For the small instances, the use of CF2, ICF2, and CF3 leads to optimal solutions for all single-platform instances, which is not true for the multiple-platforms ones. Likewise, for these instances, the solver performs better with both CF1 and ICF1 in the single-platform instances. Moreover, for the large instances, both CF2 and ICF2 result in feasible solutions for all single-platform instances, while for three multiple-platforms instances no solution is found.

Therefore, in addition to promoting the best overall performance, CF3 is the least sensitive to the existence of multiple platforms. For medium-sized instances, the average gaps for CF2 and ICF2 are 8.48% and 6.05% worse in the multiple-platforms instances than in the single-platform ones, while for CF3 this number is only 5.55%. Additionally, CF3 yields feasible solutions for the large multiple-platforms instances that CF2 and ICF2 do not.

Size	$\|\mathcal{P}\|$	Metric	BKS	CF1	ICF1	CF2	ICF2	CF3
Small	Single (8 insts.)	LB	69,840	64,636	71,576	73,071	73,071	73,071
		UB	73,071	74,274	73,165	73,071	73,071	73,071
		Gap (%)	4.06	12.44	2.00	0.00	0.00	0.00
		# of optimals	0	0	2	8	8	8
		# no feasible sol.	0	0	0	0	0	0
	Multiple (63 insts.)	LB	717	673	762	785	785	786
		UB	790	813	793	790	790	790
		Gap (%)	8.25	14.33	3.15	0.43	0.40	0.37
		# of optimals	6	20	31	48	50	51
		# no feasible sol.	0	0	0	0	0	0
Medium	Single (16 insts.)	LB	107,320	86,878	103,429	112,188	112,531	112,606
		UB	116,324	N/A	N/A	117,915	117,455	117,277
		Gap (%)	7.73	N/A	N/A	4.83	4.16	3.99
		# of optimals	0	0	0	0	0	0
		# no feasible sol.	0	5	12	0	0	0
	Multiple (12 insts.)	LB	1,224	872	1,089	1,287	1,300	1,315
		UB	1,400	N/A	N/A	1,499	1,460	1,460
		Gap (%)	12.73	N/A	N/A	13.31	10.21	9.54
		# of optimals	0	0	0	0	0	0
		# no feasible sol.	0	10	12	0	0	0
Large	Single (20 insts.)	LB	212,188	173,755	197,485	217,353	217,950	221,762
		UB	234,320	N/A	N/A	273,690	258,576	244,036
		Gap (%)	9.28	N/A	N/A	17.49	14.16	8.80
		# of optimals	0	0	0	0	0	0
		# no feasible sol.	0	17	20	0	0	0
	Multiple (12 insts.)	LB	1,618	1,194	1,437	1,612	1,617	1,626
		UB	1,841	N/A	N/A	N/A	N/A	2,050
		Gap (%)	12.15	N/A	N/A	N/A	N/A	20.74
		# of optimals	0	0	0	0	0	0
		# no feasible sol.	0	12	12	3	3	0
All insts.	Single (44 insts.)	LB	148,173	122,324	140,390	152,878	153,274	155,034
		UB	162,094	N/A	N/A	180,568	173,531	166,857
		Gap (%)	7.77	N/A	N/A	9.71	7.95	5.45
		# of optimals	0	0	2	8	8	8
		# no feasible sol.	0	22	32	0	0	0
	Multiple (87 insts.)	LB	912	772	900	968	971	975
		UB	1,019	N/A	N/A	N/A	N/A	1,056
		Gap (%)	9.41	N/A	N/A	N/A	N/A	4.45
		# of optimals	6	20	31	48	50	51
		# no feasible sol.	0	22	24	3	3	0

Table 2.3: The impact of the number of potential platforms on the CFs performances.

2.5.3 Experiments with VIs

We ran experiments with the VIs to assess their impact on the solver performance. These experiments were performed with formulations CF1, ICF1, ICF2, and CF3. The impact of VIs in CF2 was not assessed because this formulation is very similar to ICF2. Since the impacts of the VIs on the performances of the formulations is highly dependent on the instances sizes, the

results are all presented aggregated by size, not the overall average.

The experiments consisted of grouping the VIs based on similar characteristics. First, the performances were evaluated including all VIs. Then, each VI group was removed to evaluate how it affected the performance, resulting in six different configurations. Table 2.4 presents which VIs are included in each configuration. Note that ICF1 does not consider configuration VI2, since variables w are not defined in this formulation and hence configurations VI2 and All would be the same. Also, the formulations with two-index arc variables (ICF2 and CF3) do not consider symmetry breaking constraints and, thus, do not have configuration VI5. The results of the VI experiments are summarized in Tables 2.5 to 2.8.

Configuration	Meaning	CF1 VIs	ICF1 VIs	ICF2 VIs	CF3 VIs
VI1	All VIs except for the lower bounds on the number of facilities	(2.27)–(2.28) (2.30)–(2.37)	(2.27)–(2.28) (2.30)–(2.35)	(2.36)–(2.37) (2.50)–(2.57)	(2.50)–(2.55) (2.64)–(2.65)
VI2	All VIs except for the ones that relate assignment and opening of facilities	(2.27)–(2.35)	–	(2.29), (2.50)–(2.55)	(2.29), (2.50)–(2.55)
VI3	All VIs except for the lower bounds on the number of vehicles	(2.27)–(2.29) (2.31)–(2.37)	(2.27)–(2.29) (2.31)–(2.35)	(2.29), (2.36)–(2.37) (2.51)–(2.57)	(2.29), (2.51)–(2.55) (2.64)–(2.65)
VI4	All VIs except for those that relate the opening of facilities with their visit	(2.27)–(2.30) (2.36)–(2.37)	(2.27)–(2.30)	(2.29), (2.36)–(2.37) (2.50), (2.56)–(2.57)	(2.29), (2.50) (2.64)–(2.65)
VI5	All VIs except for those that break symmetry	(2.29)–(2.37)	(2.29)–(2.35)	–	–
All	All VIs	(2.27)–(2.37)	(2.27)–(2.35)	(2.29), (2.36)–(2.37) (2.50)–(2.57)	(2.29), (2.50)–(2.55) (2.64)–(2.65)

Table 2.4: Different VI configurations for each formulation.

Table 2.5 presents the results for CF1 and the six VI configurations. Only the results for the small instances are shown, since for medium and large instances no configuration of CF1 is able to find feasible solutions for all instances. Moreover, the numbers of general and binary variables are not included in this table since they do not change when including or removing

VIs.

	CF1	CF1–VI1	CF1–VI2	CF1–VI3	CF1–VI4	CF1–VI5	CF1–All
LB	7,880	8,152	8,183	7,895	8,185	8,259	8,187
UB	9,090	8,969	N/A	N/A	N/A	8,980	N/A
Gap (%)	14.11	10.73	N/A	N/A	N/A	7.44	N/A
Time (s)	2,651	2,626	2,594	2,664	2,598	2,379	2,587
# of optimals	20	21	22	20	22	27	22
# no feasible sol.	0	0	3	1	2	0	1
# of BKLBs improved	20	24	29	17	22	40	27
# of BKUBs found	34	40	38	32	35	44	37
# of constrains	4,083	4,873	4,734	4,873	4,560	4,543	4,875

Table 2.5: The impact of including VIs in CF1 for small instances (71 instances).

It is clear that the inclusion of all VIs is not beneficial for CF1 because the solver cannot find feasible solutions for all instances. Indeed, the only cases in which the inclusion of VIs is beneficial are the ones that do not include the lower bound on the number of facilities (2.29) or the symmetry breaking constraints (2.27)–(2.28). The best performing VI configuration is CF1–VI5, i.e., the one that includes all VIs except for the symmetry breaking ones. Compared to CF1 without VIs, this CF has 11.27% more constraints, leading to 4.81% LB improvement, 1.21% UB reduction, and 6.67% decrease in the average gap. Moreover, the number of instances proved optimal increased from 20 to 27, a 35% improvement. Configuration CF1–VI5 doubles the number of BKLBs improved and increases the number of BKUBs found in 29.41% compared to CF1. These improvements, however, do not get to the quality of ICF1, which outperformed them even without VIs.

Table 2.6 presents the results for the ICF1. The results show that only ICF1–VI5 (without the symmetry breaking constraints) has better results than ICF1 and showing a limited improvement. The LB increases 0.33%, the UB decreases 0.10%, the gap reduces 0.12%, the average runtime is 2.96% smaller, and two new instances have their solutions proved optimal, while four new instances have their BKUBs found.

	ICF1	ICF1–VI1	ICF1–VI3	ICF1–VI4	ICF1–VI5	ICF1–All
LB	8,741	8,675	8,645	8,729	8,770	8,684
UB	8,947	N/A	N/A	N/A	8,938	N/A
Gap (%)	3.02	N/A	N/A	N/A	2.90	N/A
Time (s)	2,097	2,397	2,349	2,317	2,035	2,352
# of optimals	33	30	31	30	35	29
# no feasible sol.	0	4	3	2	0	3
# of BKLBs improved	63	62	60	61	63	63
# of BKUBs found	51	45	45	45	55	42
# of constrains	1,319	1,968	1,968	1,655	1,639	1,970

Table 2.6: The impact of including VIs in ICF1 for small instances (71 instances).

Table 2.7 presents the results for the inclusion of VIs with ICF2 for small and medium instances. The large instances are not presented since neither the base formulation nor any of

the VI scenarios found results for all instances.

For ICF2, the inclusion of VIs is overall beneficial. For the small instances, the average LB and UB do not vary significantly. The best gaps come from ICF2–VI4 and ICF2–All, and ICF2–VI4 results in proved optimal solutions to most instances (59 against 58 from the other approaches). The running times do not vary much, even though the VIs help improving them on average. For the medium-sized instances, the best UB is achieved in configuration ICF2–VI3, the best LB in ICF2–VI2, and the best gap in ICF2–All. The number of constraints in ICF2–All increases in 4.49% with respect to ICF2, but this clearly pays off.

Size	Metric	ICF2	ICF2–VI1	ICF2–VI2	ICF2–VI3	ICF2–VI4	ICF2–All
Small (71 insts.)	LB	8,930	8,930	8,930	8,930	8,930	8,931
	UB	8,934	8,934	8,934	8,934	8,934	8,934
	Gap (%)	0.36	0.37	0.35	0.35	0.33	0.33
	Time (s)	911	843	876	823	845	851
	# of optimals	58	58	58	58	59	58
	# no feasible sol.	0	0	0	0	0	0
	# of BKLBs improved	65	65	65	65	65	65
	# of BKUBs found	69	68	70	69	70	69
	# of constraints	3,842	4,321	4,161	4,321	4,008	4,323
Medium (28 insts.)	LB	64,860	65,068	65,202	65,019	65,004	65,104
	UB	67,743	67,693	67,603	67,343	67,720	67,476
	Gap (%)	6.75	7.08	6.84	6.66	6.54	6.21
	Time (s)	3,600	3,600	3,600	3,600	3,600	3,600
	# of optimals	0	0	0	0	0	0
	# no feasible sol.	0	0	0	0	0	0
	# of BKLBs improved	27	25	28	28	27	28
	# of BKUBs found	4	4	4	6	5	3
	# of constraints	34,071	35,602	35,091	35,602	34,588	35,604

Table 2.7: The impact of including VIs in ICF2 for small and medium instances.

It is worth noticing that, with the inclusion of VIs, the performance related to ICF2–All is better than that of CF3 (the best performing formulation so far) for the medium-sized instances, improving the LB in 0.30%, the UB in 0.24%, and the gap in 0.16%. For the small instances, ICF2–All matches CF3 in LB, UB, and gap, but loses in the number of instances with optimality proved. ICF2–VI4, however, lead to the same number of instances with proved optimal solution as CF3.

Finally, Table 2.8 presents the results regarding CF3 and the different VI configurations. For the small instances, the average LB and UB do not change significantly. However, considering the optimality gap and the number of instances with proved optimal solution, the best configuration is CF3–VI1, which shows an improvement of 0.06% in the gap and provides proved optimal solutions for two extra instances compared to CF3.

For the medium-sized instances, the inclusion of VIs is mainly beneficial. The LB is improved in every configuration compared to the base formulation CF3 and the UB is improved in most of them. The best LB and UB are obtained from the inclusion of all VIs (CF3–All), improving the LB in 0.47% and the UB in 0.34% compared to CF3. CF3–All also outperforms

Size	Metric	CF3	CF3–VI1	CF3–VI2	CF3–VI3	CF3–VI4	CF3–All
Small (71 insts.)	LB	8,931	8,931	8,930	8,930	8,931	8,931
	UB	8,934	8,934	8,934	8,934	8,934	8,934
	Gap (%)	0.33	0.27	0.34	0.33	0.32	0.31
	Time (s)	861	834	820	834	834	830
	# of optimals	59	61	58	58	59	58
	# no feasible sol.	0	0	0	0	0	0
	# of BKLBs improved	65	65	65	65	65	65
	# of BKUBs found	70	70	70	69	71	70
	# of constraints	1,250	1,824	1,569	1,824	1,510	1,826
Medium (28 insts.)	LB	64,910	65,050	65,136	65,087	65,210	65,219
	UB	67,641	67,532	67,513	67,647	67,658	67,412
	Gap (%)	6.37	6.32	5.57	6.17	6.14	6.10
	Time (s)	3,600	3,600	3,600	3,600	3,600	3,600
	# of optimals	0	0	0	0	0	0
	# no feasible sol.	0	0	0	0	0	0
	# of BKLBs improved	28	28	28	28	28	28
	# of BKUBs found	3	8	3	6	6	5
	# of constraints	7,839	9,734	8,858	9,734	8,720	9,736
Large (32 insts.)	LB	139,211	139,062	138,903	138,746	139,197	139,074
	UB	153,291	N/A	N/A	154,833	155,202	N/A
	Gap (%)	13.28	N/A	N/A	14.11	14.81	N/A
	Time (s)	3,600	3,600	3,600	3,600	3,600	3,600
	# of optimals	0	0	0	0	0	0
	# no feasible sol.	0	1	2	0	0	2
	# of BKLBs improved	27	29	31	30	28	31
	# of BKUBs found	0	0	0	0	0	0
	# of constraints	27,987	32,581	30,413	32,581	30,161	32,583

Table 2.8: The impact of including VIs in CF3.

ICF2–All for these instances, improving the LB in 0.18%, the UB in 0.09%, and the gap in 0.11%. Thus, the 24.20% increase in the number of constraints compared to CF3 is worth it for these instances.

Nonetheless, for the large instances, the inclusion of VIs worsens the performance of CF3. Indeed, with CF3–VI1, CF3–VI2, and CF3–All it is not possible to find feasible solutions for some instances. Furthermore, in the configurations that do find feasible solutions for all instances (CF3–VI3 and CF3–VI4), the average LB, UB, and gap are worse than the corresponding values for CF3. Possibly this happens because these models already have large numbers of variables and constraints due to the instance size, and the inclusion of these VIs make it even harder for the MIP solver to process the branch-and-bound nodes, on top of possible effects on the solver heuristics. It is worth noticing, however, that the inclusion of VIs in CF3 for large instances allows for the improvement of another four BKLBs.

In conclusion, the effect of including valid inequalities heavily depends on the instance size and the base formulation. For the formulations with vehicle index variables (CF1 and ICF1), the inclusion of VIs helped the performance of the solver depending on which VIs were included, since in some configurations they prevented the solver from finding feasible solutions to some

instances. For the two-index arc variables formulations (ICF2 and CF3), the inclusion of VIs was beneficial for the small- and medium-sized instances. In fact, for the medium-sized instances, the best performing approach for both formulations was to include all of the VIs that were compatible with the corresponding formulation. For the large instances, however, the inclusion of VIs had negative effects in the solution quality in all evaluated scenarios.

Finally, considering all experiments performed, we have found lower bounds that are better than the BKLBs reported in the literature for 125 out of the 131 benchmark instances evaluated, which encompasses all instances with unknown optimal solutions in the literature so far. Furthermore, we have proved optimality for 55 instances for the first time. The detailed results are available in the supplementary material.

2.6 Conclusion

In this paper, we have compared mixed-integer programming (MIP) compact formulations for the two-echelon location-routing problem (2E-LRP). We have discussed a formulation with vehicle index variables from the literature and provided improvements to it. Additionally, we have introduced two novel formulations based on two-index arc variables. From a theoretical perspective, we have demonstrated that the formulations with two-index variables have stronger linear programming relaxations. We have also showed, from extensive computational experiments, that these formulations perform much better in practice when solved with a general-purposed MIP solver.

This suggests that, although the literature on the 2E-LRP is mostly based on compact formulations with a vehicle index, the future use of two-index variables formulations would be beneficial both for defining variants and evaluating the performance of tailored algorithms. Furthermore, for ad hoc methods based on mathematical formulations such as branch-and-cut schemes, decomposition-based algorithms, and matheuristics, the formulations with two-index arc variables are likely to be a better starting point than the formulations based on variables with a vehicle index.

We have also discussed the impacts of including valid inequalities in these formulations, both novel and literature-based. Our experiments suggest that their utility depends on the instance size and type of formulation. On the one hand, for small and medium instances (up to 75 customers) they help the MIP solver. On the other hand, for large instances, they actually worsen the solver performance.

Considering all experiments performed, we have improved the best known lower bounds for 125 out of the 131 benchmark instances evaluated (the other six had the optimal solution as lower bound). We have also obtained the optimal solutions of 55 instances for the first time.

Interesting research developments are available for future work. For instance, one may focus on extending the addressed formulations to the numerous 2E-LRP variants present in the literature. Moreover, the development of branch-and-cut schemes and other ad hoc solution methods

on top of these formulations could further improve the best known lower and upper bounds for these instances.

3 An exact method for a last-mile delivery routing problem with multiple deliverymen

Abstract

The demand for efficient last-mile delivery systems in large cities creates an opportunity to develop innovative logistics schemes. In this paper, we study a problem in which each vehicle may travel with more than one deliveryman to serve multiple customers with a single stop of the vehicle, increasing the delivery efficiency. We extend the vehicle routing problem with time windows and multiple deliverymen by explicitly considering the deliveryman routes. We introduce the problem, formally define it with a formulation, propose valid inequalities, and develop a tailored branch-and-Benders-cut (BBC) algorithm to solve it. The BBC is capable of solving 89% of the instances to proven optimality in reasonable times, many of them of realistic sizes. Additionally, we show the benefits of evaluating the deliveryman routes considering a cost minimization perspective, and discuss relevant solutions for urban logistics problems that can help decrease congestion and emissions.[1]

3.1 Introduction

The increasing demand for cost- and time-efficient delivery in densely populated urban areas creates additional challenges for last-mile delivery systems, such as poor traffic conditions and difficulty in finding parking locations (Martinez-Sykora et al., 2020; Boysen; Fedtke; Schwerdfeger, 2021; Reed; Campbell; Thomas, 2024). However, the proximity of customers allows for inventive developments to overcome these challenges. For instance, the combination of trucks and drones is already well-known (Li et al., 2021a) since the seminal work by Murray and Chu (2015). Similarly, the combination of robots and trucks has also been applied to last-mile delivery systems (Alfandari; Ljubić; De Melo da Silva, 2022). Alternatively, one could rely on

[1] This chapter is a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos). It has been published at the European Journal of Operational Research (Senna et al., 2024a).

crowd-sourcing operations in last-mile delivery, as proposed by Ouyang, Leung, and Huang (2023), or on combining vehicles, cargo bikes, and walking porters, such as in the problem presented by Bayliss et al. (2023).

Another well-adopted possibility in city logistics is the combination of vehicles with walking carriers (Reed; Campbell; Thomas, 2021; Wehbi; Bektaş; Iris, 2022; Le Colleter et al., 2023). In particular, Pureza, Morabito, and Reimann (2012) proposed the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD), which arose from a practical application of last-mile delivery from a beverage company. In this problem, a vehicle may travel with more than one deliveryman. Once the vehicle parks, the deliverymen walk to serve the customers in parallel. This reduces the time the vehicle stays parked throughout the route, allowing it to serve more customers in a single route. Therefore, a smaller fleet of vehicles can serve the same customers compared to the traditional approach of having a single deliveryman traveling in each vehicle. Since deliverymen fixed costs are smaller than those of the vehicles, this creates an opportunity for operational cost reduction.

The VRPTWMD is often modeled using a network given by nodes that correspond to clusters of customers (Pureza; Morabito; Reimann, 2012; Álvarez; Munari, 2017; Munari; Morabito, 2018). Clusters are defined in advance, in a previous decision stage, and the service time at a cluster depends on the number of deliverymen in the vehicle that visits that cluster. Hence, at each stop of a vehicle at a node, the service time at this node is the service time of the cluster divided by the number of deliverymen on the vehicle. Some variants consider the definition of the clusters as an endogenous decision, thus determining also the clustering of customers that are visited at each stop of the vehicles, as in Senarclens de Grancy and Reimann (2015). However, in these variants, the authors still simply divide the service time of a cluster by the number of deliverymen that serve it. To the best of our knowledge, no study has addressed the VRPTWMD and related variants explicitly considering the routes traveled by the deliverymen inside the clusters. Moreover, authors have assumed thus far that the deliverymen capacities are small compared to the customer demands, such as in the beverage industry from which the problem emerged, making the deliveryman routes trivial. However, in applications where the customer demands are small (e.g., e-commerce) or the deliverymen capacities are large (e.g., deliverymen with small carts or cargo bikes), this assumption is not valid and the deliveryman routes can significantly affect the vehicle routes.

In this paper, we extend the VRPTWMD by also designing the deliveryman routes inside each cluster, instead of simply considering round-trips. Since most drone-truck and robot-truck combinations consider that drones and robots can only visit one customer at a time (Moshref-Javadi; Winkenbach, 2021; Ostermeier; Heimfarth; Hübner, 2023), our work also generalizes such problems. Furthermore, to efficiently solve the problem, we propose a Benders decomposition-based exact algorithm (Benders, 1962), which might be of broader interest given that the majority of works that address the VRPTWMD and related problems rely on heuristics (Pureza; Morabito; Reimann, 2012; Senarclens de Grancy; Reimann, 2014; Moshref-Javadi;

Winkenbach, 2021; Wehbi; Bektaş; Iris, 2022; Le Colleter et al., 2023).

The contributions of this paper are threefold. First, we introduce a novel problem in the literature with practical and theoretical relevance, namely the vehicle routing problem with time windows, multiple deliverymen, and two-level routing (VRPTWMD2R). Second, we present a formulation for this problem and introduce several families of valid inequalities that tighten the linear programming (LP) relaxation of this formulation. Third, we propose a branch-and-Benders-cut method to solve the problem, which is an exact algorithm based on Benders decomposition, and develop lower bounding techniques.

The remainder of this paper is organized as follows. Section 3.2 reviews the pertinent literature. In Section 3.3, the problem is defined. Section 3.4 introduces the mathematical formulation and valid inequalities. Section 3.5 describes the exact algorithm to solve the problem. In Section 3.6, the computational experiments are outlined and the results are evaluated. Finally, Section 3.7 presents concluding remarks.

3.2 Literature review

Pureza, Morabito, and Reimann (2012) introduced the VRPTWMD as a variant of the classical vehicle routing problem (VRP). In this variant, in addition to time windows and vehicle capacity constraints, the vehicles may carry more than one deliveryman to reduce overall service time. The problem arises from companies that make regular deliveries in densely populated urban areas, in which the proximity of customers creates the possibility of serving more than one customer with a single stop of the vehicle. In such case, the presence of multiple deliverymen allows the customers to be served in parallel, reducing the time of each stop of the vehicles. Since the vehicle fixed costs are usually higher than those of the deliverymen, increasing the number of deliverymen can reduce the number of vehicles needed, decreasing the overall costs.

The problem dynamics are based on the creation of clusters of customers with similar time windows and close to each other. The vehicles travel from the depot to the clusters and, once they arrive, the deliverymen leave the vehicle to serve the customers. Once all customers in a cluster are served, the deliverymen return to the vehicle and travel to the next cluster on the vehicle route.

Several authors have studied this problem with different approaches. Pureza, Morabito, and Reimann (2012) compared the performance of two metaheuristics: tabu search (TS) and ant colony optimization (ACO). Senarclens de Grancy and Reimann (2014) systematically compared the performance of ACO and greedy randomized adaptative search procedure (GRASP) to solve the problem. Álvarez and Munari (2016) solved the problem with iterated local search (ILS) and large neighborhood search (LNS). Munari and Morabito (2018) proposed the first exact algorithm for the problem, which consisted of a branch-price-and-cut method, thus based on the column generation technique. Álvarez and Munari (2017) combined this exact method with the metaheuristics ILS and LNS, resulting in a hybrid method for the problem. Souza Neto and

Pureza (2016) proposed a variant of the VRPTWMD in which vehicles can perform more than one route and solved it with GRASP, a commercial solver, and a hybrid method.

All of the above-mentioned studies address the problem considering two simplifying hypotheses: (*i*) the clusters are predefined, and (*ii*) the time spent in each cluster is approximated by a function of the cluster demand and the number of deliverymen, ignoring the routes traveled by the deliverymen. To incorporate clustering issues, Senarclens de Grancy and Reimann (2015) proposed two heuristics to cluster the customers, and Senarclens de Grancy (2015) combined these heuristics in an iterative method to optimize clustering and routing.

We are not aware of any study addressing the design of deliveryman routes within the VRPTWMD. Approximating the service time of clusters based on their demand and the number of deliverymen may be reasonable when the deliverymen capacities are small compared to the customers demands. In such cases, the deliverymen can only visit one customer in each of their routes, making the optimal deliveryman routes trivial (i.e., round trips), with no need to be optimized. However, when the deliverymen capacities are large compared to the customers demands, they can visit more than one customer in each route. In such cases, approximating the cluster service time based on the demand and the number of deliverymen becomes less accurate and does not represent the problem complexity. This assessment is important because it affects all of the other decisions of the problem, namely the number of vehicles and deliverymen, and the vehicle routes.

Another issue with disregarding the deliveryman routes arises when considering time windows. Preprocessing the service times in each cluster, such as made by Pureza, Morabito, and Reimann (2012), implicitly defines the deliveryman routes in advance. To accomodate this, the time windows for parking locations need to be adapted to ensure that the vehicle arrives at each cluster with enough time to serve the customers in this predetermined order while respecting their time windows. The flexibility created by jointly optimizing the deliveryman routes allows for the adoption of deliveryman routes that can still respect customers' time windows if the vehicle arrives at the cluster later than it would if these routes were predefined. This way, there is no need to adapt the time windows of parking locations, making the problem less constrained and creating opportunity for cost reduction. Finally, there is a trade-off between the time spent on a cluster and the cost of the deliveryman routes within this cluster. Cost and time minimization not always lead to the same solution. If these routes are preprocessed, a subset of them must be chosen a priori and there is no guarantee that the selected ones will lead to the best overall solution.

The present study addresses this issue by generalizing the VRPTWMD to consider two-level routing (VRPTWMD2R), i.e., both the vehicle and the deliveryman routes. Therefore, while relying on assumption (*i*), i.e., the clusters of customers are previously defined, we propose a problem that overcomes the limitations of assumption (*ii*) by properly evaluating and optimizing the deliveryman routes.

3.3 Problem definition

We define the VRPTWMD2R considering different graph representations for vehicle and deliveryman routes (first- and second-level). For vehicle routes, we assume a single depot and a set of clusters $N = \{1, 2, \ldots, n\}$, where $n > 0$ is the number of clusters. Each cluster consists of a set of customers and a single parking location. We shall refer to clusters and parking locations interchangeably. Let $G = (N_0, A)$ be a directed graph, where $N_0 = \{0, n+1\} \cup N$ is the set of nodes and $A = \{(i,j) \mid i,j \in N_0, i \neq j,\ i \neq n+1,\ j \neq 0\}$ is the set of arcs. Indices 0 and $n+1$ represent the depot, and all vehicle routes start at 0 and end at $n+1$. This graph only concerns the nodes and arcs related to the design of first-level routes.

For each cluster $i \in N$, we define a directed graph $G^i = (N_0^i, A^i)$, given by the set of nodes $N_0^i = \{0_i, n_i + 1\} \cup N^i$, where N^i is the set of n_i customer nodes in this cluster and $A^i = \{(h,k) \mid h,k \in N_0^i, h \neq k,\ h \neq n_i + 1,\ k \neq 0_i\}$ is the set of arcs related to the second-level routes inside this cluster. Nodes 0_i and $n_i + 1$ represent the parking location, and the deliveryman routes must depart from 0_i and return to $n_i + 1$, traversing only the arcs in A^i. Both nodes 0_i and $n_i + 1$ are at the same place as the corresponding parking location $i \in N$. No customer is part of more than one cluster, i.e., $N^i \cap N^j = \emptyset, \forall\ i,j \in N, i \neq j$. To make the notation clear, we shall represent nodes of first-level routes (set N) by i and j, and those of second-level routes (sets $N^i, i \in N$) by h and k.

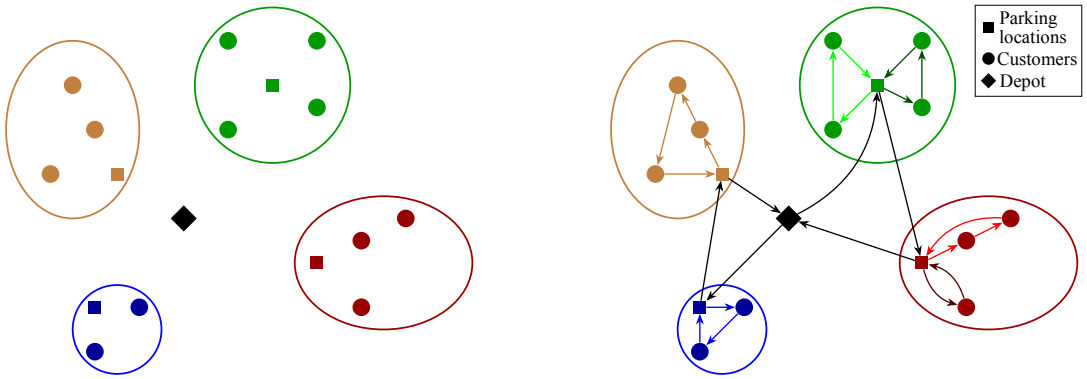
Every cluster is served by exactly one vehicle, and every customer inside a cluster is served by exactly one deliveryman. Both clusters and customers have time windows that indicate when the service may begin, which are supposed to be compatible in order to ensure feasibility. Customers have positive demands that are aggregated to define cluster demands, typically consisting of a few customers. We assume that deliverymen do not have capacity constraints since clusters are relatively small, and hence all customers of a cluster could be served by a single deliveryman when considering only capacity constraints.

Each vehicle may travel with up to M_L deliverymen. Once the vehicle arrives at a cluster, the deliverymen leave it to serve the customers. After serving all of them, the deliverymen return to the vehicle and it travels to the next cluster in the route. We define the set of possible numbers of deliverymen in a vehicle as $L = \{1, 2, \ldots, M_L\}$. We assume that the vehicle fleet and the deliveryman team are both homogeneous.

The decisions of the problem are (*i*) the number of vehicles to be used, (*ii*) the number of deliverymen in each vehicle, (*iii*) the vehicle routes, and (*iv*) the deliveryman routes. These decisions should be made ensuring that every customer is served, and respecting time windows and vehicle capacity. The goal of the problem is to minimize the fixed costs associated with vehicles and deliverymen and the distance-related costs of vehicle and deliveryman routes.

Figure 3.1 illustrates the VRPTWMD2R. Figure 3.1a presents an instance of the problem, with the customers clustered around their respective parking locations. Figure 3.1b represents a feasible solution to the problem. The black arrows that travel among parking locations represent

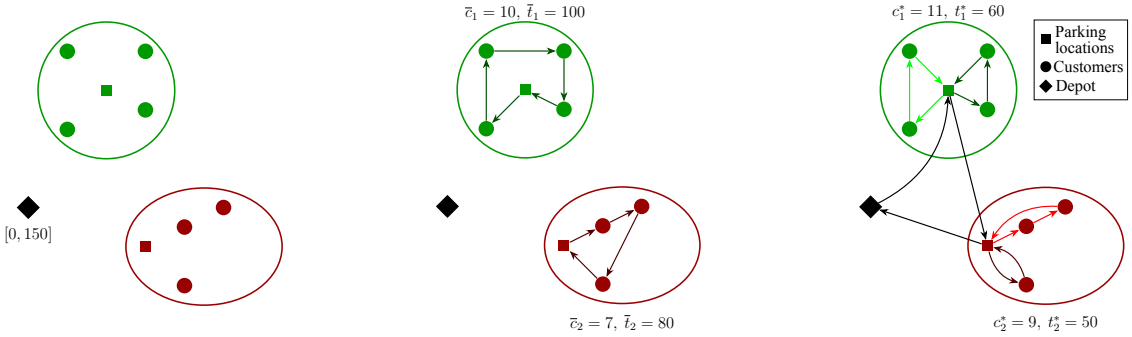
vehicle paths, while the colored arrows inside the clusters show deliveryman routes. In the picture, the vehicle that serves the clusters on the left-hand side of the picture travels with one deliveryman and the other one travels with two deliverymen.



(a) Instance. (b) Solution.

Figure 3.1: An illustrative example of the VRPTWMD2R.

A trade-off between vehicle and deliveryman costs is inherent to the VRPTWMD2R. Figure 3.2 illustrates it. Figure 3.2a presents an instance of the problem in which the depot time window closes at instant 150. The best solution considering only the second-level routes cost minimization would be serving each cluster with a single deliveryman, as portrayed in Figure 3.2b. This solution incurs in costs $\bar{c}_1 = 10$ and $\bar{c}_2 = 7$, while the time spent in each cluster is $\bar{t}_1 = 100$ and $\bar{t}_2 = 80$. If these routes were to be taken, these clusters would need to be served by two vehicles since they would not respect the depot time windows when served by a single vehicle. However, if the problem is solved by minimizing all costs, the solution would be the one represented in Figure 3.2c, in which two deliverymen travel with a single vehicle. The routes inside the clusters are slightly more costly when considered individually and include an additional deliveryman, but they help minimize the overall costs.



(a) Instance. (b) Min deliveryman routes costs. (c) Min overall costs.

Figure 3.2: A trade-off between deliveryman routes cost and time.

3.4 Mathematical formulation

We introduce a compact mixed-integer programming (MIP) formulation for the VRPTWMD2R. Consider the following parameters:

M_L Maximum number of deliverymen in each vehicle;

f_v Fixed cost associated with each vehicle;

f_d Fixed cost associated with each deliveryman;

c_v Unitary distance cost of first-level routes (vehicles);

c_d Unitary distance cost of second-level routes (deliverymen);

Q Vehicle load capacity;

q_i Demand of cluster $i \in N$;

d_{ij} Distance between first-level nodes i and $j, (i, j) \in A$ (asymmetrical);

t_{ij} Travel time between first-level nodes i and $j, (i, j) \in A$ (asymmetrical);

d^i_{hk} Distance between second-level nodes h and k of cluster $i \in N, (h, k) \in A^i$, (asymmetrical);

t^i_{hk} Travel time between second-level nodes h and k of cluster $i \in N, (h, k) \in A^i$ (asymmetrical);

s_h Service time of customer $h \in N^i$ of cluster $i \in N$;

$[a_h, b_h]$ Time window of node $h \in N^i_0$ of cluster $i \in N$.

We define the decision variables taking into account the first- and second-level routes, related to vehicles and deliverymen. Additionally, we need auxiliary variables to model vehicle load and time propagation in the routes. These variables are defined as follows:

x_{ijl} Binary variable that indicates whether a vehicle travels from node i to node j with l deliverymen in a first-level route, $(i, j) \in A, l \in L$;

u_i Vehicle load after leaving node $i \in N_0$;

x^i_{hk} Binary variable that indicates whether a deliveryman travels from node h to node k in a second-level route inside cluster i, $(h, k) \in A^i, i \in N$;

w_h Time when service at node $h \in N^i_0, i \in N$, begins. The arrival time of the vehicle at the parking location of cluster i is represented by w_{0_i} and its departure happens at w_{n_i+1}.

Using the sets, parameters, and decision variables defined so far, we propose the following compact formulation (CF) for the VRPTWMD2R:

$$\text{(CF) min} \sum_{j \in N} \sum_{l \in L} (f_v + l f_d) x_{0jl} + c_v \sum_{(i,j) \in A} \sum_{l \in L} d_{ij} x_{ijl} + c_d \sum_{i \in N} \sum_{(h,k) \in A^i} d^i_{hk} x^i_{hk} \tag{3.1}$$

$$\text{s.t.} \sum_{i:(i,j)\in A} \sum_{l\in L} x_{ijl} = 1,\ \forall\, j \in N \tag{3.2}$$

$$\sum_{i:(i,j)\in A} x_{ijl} = \sum_{i:(j,i)\in A} x_{jil},\ \forall\, j \in N, l \in L \tag{3.3}$$

$$\sum_{i\in N} x_{0il} = \sum_{i\in N} x_{i(n+1)l},\ \forall\, l \in L \tag{3.4}$$

$$u_j \geq u_i + q_j - Q\left(1 - \sum_{l\in L} x_{ijl}\right),\ \forall\, (i,j) \in A \tag{3.5}$$

$$\sum_{h:(h,k)\in A^i} x^i_{hk} = 1,\ \forall\, k \in N^i, i \in N \tag{3.6}$$

$$\sum_{h:(h,k)\in A^i} x^i_{hk} = \sum_{h:(k,h)\in A^i} x^i_{kh},\ \forall\, k \in N^i, i \in N \tag{3.7}$$

$$\sum_{h\in N^i} x^i_{0_i h} = \sum_{h\in N^i} x^i_{h(n_i+1)},\ \forall\, i \in N \tag{3.8}$$

$$w_k \geq w_h + s_h + t^i_{hk} - M^i_{hk}(1 - x^i_{hk}),\ \forall\, (h,k) \in A^i, i \in N \tag{3.9}$$

$$w_{0_j} \geq w_{n_i+1} + t_{ij} - M_{ij}\left(1 - \sum_{l\in L} x_{ijl}\right),\ \forall\, (i,j) \in A \tag{3.10}$$

$$\sum_{h\in N^j} x^j_{0_j h} \leq \sum_{i:(i,j)\in A} \sum_{l\in L} l x_{ijl},\ \forall\, j \in N \tag{3.11}$$

$$u_0 = 0, w_0 = 0 \tag{3.12}$$

$$x_{ijl} \in \{0,1\},\ \forall\, (i,j) \in A, l \in L \tag{3.13}$$

$$q_i \leq u_i \leq Q,\ \forall\, i \in N_0 \tag{3.14}$$

$$x^i_{hk} \in \{0,1\},\ \forall\, (h,k) \in A^i, i \in N \tag{3.15}$$

$$a_h \leq w_h \leq b_h,\ \forall\, h \in N^i_0, i \in N_0. \tag{3.16}$$

The objective function (3.1) seeks to minimize the total fixed costs of both vehicles and deliverymen and the distance costs of both vehicle and deliveryman routes. Constraints (3.2) ensure that every cluster is visited by exactly one vehicle. Constraints (3.3) and (3.4) are flow conservation constraints for first-level routes. Constraints (3.5) control the load flow in vehicle routes. Constraints (3.6)–(3.8) are similar to (3.2)–(3.4) but considering second-level routes. Constraints (3.9) and (3.10) control the time propagation for deliveryman and vehicle routes, respectively. Beyond defining the arrival time at each customer, constraints (3.9) implicitly define the time spent in each cluster since they involve the moments that the deliverymen depart from and arrive at the parking locations. Constraints (3.10) use this information to synchronize the first- and second-level routes by defining that the deliverymen start to serve a cluster j after having served a cluster i and having traveled to cluster j if they travel in a vehicle that goes from i to j. In these constraints, we define $M^i_{hk} = \max\{0, b_h + s_h + t^i_{hk} - a_k\}$ and $M_{ij} = \max\{0, b_{n_i+1} + t_{ij} - a_{0_j}\}$ as the smallest possible values to ensure that the constraints

are valid. Constraints (3.11) also couple the first- and second-level routes by defining that the number of deliveryman routes inside a cluster is, at most, the number of deliverymen that arrive at it (it is possible that not all deliverymen visiting a cluster leave the vehicle). Constraints (3.12)–(3.16) define the domain of the decision variables.

3.4.1 Valid inequalities

Formulation CF can be strengthened by the following valid inequalities (VIs) to improve its linear relaxation. In these constraints, let $e_{il}, i \in N, l \in L$, be a lower bound on the time needed to serve cluster i with l deliverymen and $m_i, i \in N$, be a lower bound on the number of deliverymen needed to serve cluster i feasibly.

$$\sum_{h \in N^i} x^i_{0_i h} \geq 1, \ \forall \, i \in N \tag{3.17}$$

$$\sum_{(h,k) \in A^i : h,k \in S} x^i_{hk} \leq |S| - 1, \ \forall \, S \subset N^i, i \in N : |S| \in \{2, 3\} \tag{3.18}$$

$$x^i_{hk} = 0, \ \forall \, (h,k) \in A^i, i \in N : (a_h + s_h + t^i_{hk} > b_k) \tag{3.19}$$

$$\sum_{j \in N} \sum_{l \in L} x_{0jl} \geq \left\lceil \frac{1}{Q} \sum_{i \in N} q_i \right\rceil \tag{3.20}$$

$$\sum_{(i,j) \in A : i,j \in S} \sum_{l \in L} x_{ijl} \leq |S| - 1, \ \forall \, S \subset N : |S| \in \{2, 3\} \tag{3.21}$$

$$x_{ijl} = 0, \ \forall \, (i,j) \in A, l \in L : (q_i + q_j > Q) \vee (a_{n_i+1} + t_{ij} > b_{0_j}) \tag{3.22}$$

$$w_{n_i+1} \geq w_{0_i} + \sum_{j : (i,j) \in A} \sum_{l \in L} e_{il} x_{ijl}, \ \forall \, i \in N \tag{3.23}$$

$$x_{ijl} = 0, \ \forall \, (i,j) \in A, l \in L : (l < m_i) \vee (l < m_j) \tag{3.24}$$

$$\sum_{h \in N^i} x^i_{0_i h} \geq m_i, \ \forall \, i \in N. \tag{3.25}$$

Constraints (3.17)–(3.22) are common in the literature (Dantzig; Fulkerson; Johnson, 1954; Ascheuer; Fischetti; Grötschel, 2001; Lysgaard; Letchford; Eglese, 2004), while constraints (3.23)–(3.25) are novel VIs proposed specifically for this problem. Constraints (3.17) ensure that at least one deliveryman leaves each parking location. Constraints (3.18) eliminate small subtours of two and three customers in second-level routes. Constraints (3.19) remove infeasible second-level arcs due to time window incompatibility. Constraints (3.20) define a lower bound on the number of vehicles needed to serve all the clusters based on the total cluster demands and vehicle capacity. Constraints (3.21) eliminate subtours for sets of two and three clusters in first-level routes. Constraints (3.22) eliminate first-level arcs that are infeasible due to vehicle capacity or time windows incompatibility. Constraints (3.23) provide an estimation on the minimum time spent on the cluster. Constraints (3.24) forbid the visit of the cluster by a vehicle with fewer deliverymen than needed to serve it. Constraints (3.25) ensure that the number of

deliverymen leaving a parking location respects its lower bound. Since $m_i \geq 1,\ \forall\, i \in N$, constraints (3.17) are redundant when constraints (3.25) are considered. Hence, either constraints (3.17) or (3.25) are included, never both.

On top of these constraints, time windows are tightened based on the earliest arrival time from the depot and the latest departure time to arrive while the depot is still open (Ascheuer; Fischetti; Grötschel, 2001).

3.5 Benders decomposition

Since the definition of the deliveryman routes depends on the vehicle routes and the number of deliverymen serving each cluster, the CF can be decomposed in a Benders fashion (Benders, 1962; Hooker; Ottosson, 2003; Codato; Fischetti, 2006). This way, the master problem (MP) defines the first-level routes and the number of deliverymen in each vehicle, and the subproblem (SP) defines the second-level routes.

To exploit this characteristic of the VRPTWMD2R and efficiently solve it, we develop an exact algorithm based on a branch-and-Benders-cut (BBC) scheme (Moreno; Munari; Alem, 2019, 2020). To this extent, we improve the Benders decomposition by including valid inequalities and developing lower bounding techniques. Section 3.5.1 presents the MP, Section 3.5.2 defines the SP, Section 3.5.3 introduces useful lower bounds, and Section 3.5.4 discusses the BBC algorithm.

3.5.1 Master problem

Let $\eta_i, i \in N$, be a variable representing the cost of the deliveryman routes inside cluster i with a lower bound $\underline{\eta}_i \geq 0$. Let R be the set of all pairs (r, l) of vehicle routes r and number of deliverymen l that are feasible given first-level constraints (3.2)–(3.5), (3.10), (3.12)–(3.14) and second-level constraints (3.6)–(3.9), (3.11), (3.15), (3.16); and $\overline{R}$ be the set of pairs (r, l) that are feasible considering first-level constraints (information in the MP), but infeasible considering second-level constraints (information in the SP). It is clear that $R \cap \overline{R} = \emptyset$.

Let N_r be the set of clusters visited by route r and A_r be the set of arcs of route r. Given a pair $(r, l) \in R$, let $g_{rli}, i \in N_r$, represent the cost of deliveryman routes inside cluster i when visited by a vehicle traveling with l deliverymen along route r, and $c_{rl} = \sum_{i \in N_r} g_{rli}$ be the sum of these costs throughout the vehicle route.

Given these definitions, the CF can be reformulated as the following MP:

$$\text{(MP) min} \sum_{j \in N} \sum_{l \in L} (f_v + l f_d) x_{0jl} + c_v \sum_{(i,j) \in A} \sum_{l \in L} d_{ij} x_{ijl} + \sum_{i \in N} \eta_i \tag{3.26}$$

$$\text{s.t. (3.2)–(3.5), (3.10), (3.12)–(3.14)}$$

$$\sum_{i \in N_r} \eta_i \geq c_{rl} \left(\sum_{(i,j) \in A_r} x_{ijl} - |A_r| + 1 \right), \ \forall \, (r,l) \in R \tag{3.27}$$

$$\sum_{(i,j) \in A_r} x_{ijl} \leq |A_r| - 1, \ \forall \, (r,l) \in \overline{R} \tag{3.28}$$

$$a_h \leq w_h \leq b_h, \ \forall \, h \in \{0_i, n_i + 1\}, i \in N \tag{3.29}$$

$$\eta_i \geq \underline{\eta}_i, \ \forall \, i \in N. \tag{3.30}$$

The objective function (3.26) is equivalent to (3.1) with a different form of calculating the deliveryman routes cost. Constraints (3.27) correspond to the so-called optimality cuts, which define the cost of second-level routes inside the clusters visited by a vehicle traveling along a first-level route r and carrying l deliverymen. Constraints (3.28) consist in the so-called feasibility cuts, removing from the set of feasible solutions of the MP the vehicle routes that are infeasible due to the corresponding deliveryman routes. Constraints (3.29) define the time windows of parking locations, and constraints (3.30) establish a lower bound on the cost of deliveryman routes inside each cluster. The MP can be further strengthened by VIs (3.20)–(3.24). We shall refer to the MP without the optimality cuts (3.27) and feasibility cuts (3.28) as the relaxed MP (RMP).

Constraints (3.27) and (3.28) are based on the traditional route-based optimality and feasibility cuts. However, we propose using the path cuts introduced by Parada et al. (2024), in which the first-level route arcs that are connected to the depot are removed from the cut. Propositions 1 and 2 ensure the validity of this approach for the VRPTWMD2R. Proposition 3 includes an additional summation in $l \in L$ in the feasibility cuts. These modifications yield better cuts that help boost the algorithm's performance. To this extent, we denote by $\hat{A}_r \subset A_r$ the set of arcs in route r without those connected to the depot.

Proposition 3.1. *The constraints*

$$\sum_{i \in N_r} \eta_i \geq c_{rl} \left(\sum_{(i,j) \in \hat{A}_r} x_{ijl} - |\hat{A}_r| + 1 \right), \ \forall \, (r,l) \in R \tag{3.31}$$

can replace constraints (3.27) *as valid optimality cuts if* $|N_r| > 1$ *and the triangular inequality holds for vehicle routes.*

Proof. Given a pair $(r,l) \in R$ with $|N_r| > 1$, let $r = (0, r_1, r_2, \ldots, r_{|N_r|}, n+1)$ be the sequence of nodes visited in first-level route r. Let us define path $p = (r_1, r_2, \ldots, r_{|N_r|})$ as the path of $|N_r|$ clusters visited in route r. With these definitions, $\hat{A}_r$ can be interpreted as the set of arcs of p. Hence, constraints (3.31) state that, for every first-level route that contains path p, the cost of second-level routes inside the clusters of path p is at least c_{rl}, i.e., the cost of traveling the path in a vehicle route that does not visit any cluster out of the path. This is true because in every first-level route $\overline{r} \supset p, \overline{r} \neq r$, there are clusters visited before and/or after path p, making

the dynamic of the deliverymen inside the clusters of path p more constrained than in route r, as triangular inequality holds. Since it is more constrained, the costs of the deliveryman routes in the clusters of path p is at least c_{rl}, proving the validity of constraints (3.31) as optimality cuts. □

Proposition 3.2. *The constraints*

$$\sum_{(i,j)\in\hat{A}_r} x_{ijl} \leq |\hat{A}_r| - 1,\ \forall\,(r,l) \in \overline{R} \tag{3.32}$$

can replace constraints (3.28) *as valid feasibility cuts if the triangular inequality holds for vehicle routes.*

Proof. Following the notation used on the proof of Proposition 3.1, constraints (3.32) state that $(r,l) \in \overline{R} \Rightarrow (\overline{r},l) \in \overline{R},\ \forall\,\overline{r} \supset p$, i.e., if a first-level route $r = (0,p,n+1)$ is infeasible when traveled by a vehicle with l deliverymen, every other route $\overline{r} \supset p$ will also be infeasible when traveled with the same number l of deliverymen. This is true because, if the triangular inequality holds, including any cluster before or after path p would make the second-level routes inside the clusters of p more constrained than in route r. If these deliveryman routes are infeasible without this additional cluster, they will remain as such with this addition. □

Proposition 3.3. *The constraints*

$$\sum_{(i,j)\in\hat{A}_r} \sum_{\bar{l}\in L:\bar{l}\leq l} x_{ij\bar{l}} \leq |\hat{A}_r| - 1,\ \forall\,(r,l) \in \overline{R} \tag{3.33}$$

can replace constraints (3.28) *as valid feasibility cuts if the triangular inequality holds for vehicle routes.*

Proof. It is true that $(r,l) \in \overline{R} \Rightarrow (r,\bar{l}) \in \overline{R},\ \forall\,\bar{l} \in L, \bar{l} < l$, because reducing the number of deliverymen on a first-level route makes the second-level routes inside the clusters more constrained. Thus, if the first-level route is infeasible with l deliverymen, it will also be with $\bar{l} < l$. Therefore, given Proposition 3.2,

$$\sum_{(i,j)\in\hat{A}_r} x_{ij\bar{l}} \leq |\hat{A}_r| - 1,\ \forall\,\bar{l} \leq l, (r,l) \in \overline{R}$$

are valid feasibility cuts if the triangular inequality holds. By constraints (3.2)–(3.4), at most one value of l is associated with a vehicle route r, allowing for the summation in $\bar{l}$ that yields constraints (3.33) as valid feasibility cuts. □

Note that it is possible to aggregate the optimality cuts (3.31) by summing them up for all number of deliverymen $\bar{l} < l$, as we did for feasibility cuts (3.33). However, preliminary results indicate that, in the case of optimality cuts, this is only beneficial for small instances, and has a

negative effect for medium and large instances as the cuts become too dense. Therefore, we use the disaggregated version as presented above.

Comparing the improved path cuts (3.31) and (3.33) with the original route cuts (3.27) and (3.28), it is clear that the improved versions yield stronger LP relaxations. Furthermore, while each route cut is active in a single integer solution, the improved versions are active in more than one solution. This justifies the improvements from a theoretical perspective. Our experiments confirm that this theoretical improvement is translated into a better performance of the BBC, as shown in Section 3.6.3.

Another possible improvement to the MP is the replacement of constraints (3.5) and (3.14) by the so-called rounded capacity inequalities (RCIs):

$$\sum_{\substack{(i,j)\in A:\\ i\notin S, j\in S}} \sum_{l\in L} x_{ijl} \geq \left\lceil \frac{1}{Q} \sum_{i\in S} q_i \right\rceil, \ \forall\, S \subset N. \tag{3.34}$$

These constraints are know to have a stronger linear relaxation than constraints (3.5) and (3.14) and to be efficient in branch-and-cut algorithms for VRP variants (Lysgaard; Letchford; Eglese, 2004). Since they are exponential by definition, their inclusion is made dynamically throughout the exploration of the branch-and-cut tree whenever they are found to be violated.

3.5.2 Subproblem

To generate optimality and feasibility cuts we resort to an SP that optimizes the cost of the deliveryman routes for each pair $(r, l) \in R$, or determines that it is infeasible to perform first-level route r with l deliverymen if $(r, l) \in \overline{R}$. Given a pair (r, l), the corresponding SP is defined by

$$\text{(SP) min } c_d \sum_{i\in N_r} \sum_{(h,k)\in A^i} d^i_{hk} x^i_{hk} \tag{3.35}$$

$$\text{s.t.} \sum_{h:(h,k)\in A^i} x^i_{hk} = 1, \ \forall\, k \in N^i, i \in N_r \tag{3.36}$$

$$\sum_{h:(h,k)\in A^i} x^i_{hk} = \sum_{h:(k,h)\in A^i} x^i_{kh}, \ \forall\, k \in N^i, i \in N_r \tag{3.37}$$

$$\sum_{h\in N^i} x^i_{0_i h} = \sum_{h\in N^i} x^i_{h(n_i+1)}, \ \forall\, i \in N_r \tag{3.38}$$

$$w_k \geq w_h + s_h + t^i_{hk} - M^i_{hk}(1 - x^i_{hk}), \ \forall\, (h,k) \in A^i, i \in N_r \tag{3.39}$$

$$\sum_{h\in N^i} x^i_{0_i h} \leq l, \ \forall\, i \in N_r \tag{3.40}$$

$$w_{0_j} \geq w_{n_i+1} + t_{ij}, \ \forall\, (i,j) \in A_r \tag{3.41}$$

$$x^i_{hk} \in \{0,1\}, \ \forall\, (h,k) \in A^i, i \in N_r \tag{3.42}$$

$$a_h \leq w_h \leq b_h, \ \forall\, h \in N^i_0, i \in N_r \cup \{n+1\}. \tag{3.43}$$

The objective function (3.35) seeks to minimize the total cost of second-level routes. Constraints (3.36)–(3.39) are equivalent to (3.6)–(3.9) but restricted to the nodes in N_r. Constraints (3.40) limit the number of deliveryman routes inside a cluster to the number of deliverymen traveling in the vehicle route r. Constraints (3.41) define the vehicle time flow, i.e., the deliverymen leave a parking location (w_{0_j}) after serving the previous cluster in the route (w_{n_i+1}) and traveling from one cluster to the next one in the vehicle route (t_{ij}). Finally, constraints (3.42) and (3.43) define the domain of the decision variables.

It is important to notice that this SP comes from splitting a solution in routes and is, therefore, separable by vehicle route r, but not by deliveryman routes in each cluster due to the trade-off between deliveryman routes cost and time discussed in Section 3.3. There is a time dependency among different clusters served by the same vehicle given by constraints (3.41). Thus, although there might be a short deliveryman route to serve a given cluster's customers, if this route takes a long time it might affect the feasibility of the corresponding vehicle route by not respecting the next cluster's time window. Hence, in this case, it would be necessary to take longer deliveryman routes that would be more costly but feasible considering the vehicle route to be followed.

The SP can be strengthened by VIs (3.17)–(3.19), and (3.25). We also define the following VIs for the SP relative to a pair $(r, l) \in R \cup \overline{R}$:

$$w_{n_i+1} \geq w_{0_i} + e_{il}, \ \forall\, i \in N_r, \tag{3.44}$$

in which e_{il} is a lower bound on the time spent in cluster i when visited by a vehicle with l deliverymen, as discussed in Section 3.4.1. Thus, these constraints define that the time spent in each cluster is at least this lower bound.

Finally, time windows are tightened. Ascheuer, Fischetti, and Grötschel (2001) propose to tighten the time windows based on all of the possible predecessors and successors of a node. Since in the SP the vehicle route is predefined, each cluster has a unique predecessor and a unique successor, making this tightening very efficient.

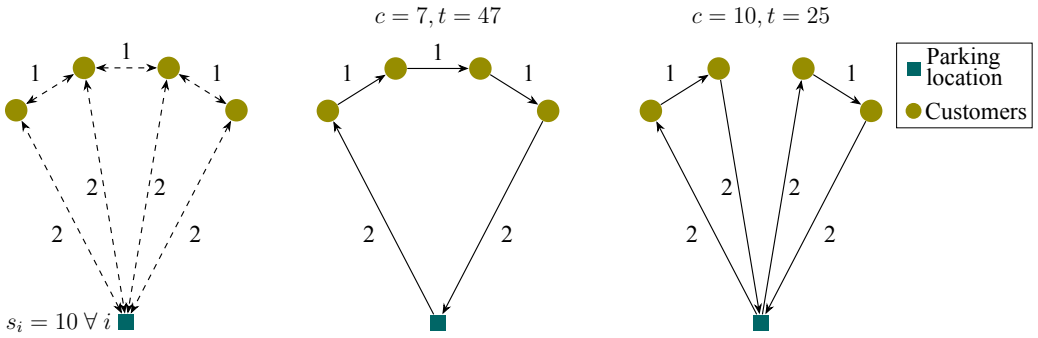
3.5.3 Lower bounds

The definition of the MP relies on the lower bound $\underline{\eta}_i, i \in N$, for the cost of the deliveryman routes inside a cluster i. Moreover, VIs (3.23) and (3.44) depend on the lower bound $e_{il}, i \in N, l \in L$, for the time spent in cluster i when served with l deliverymen; and VIs (3.24) and (3.25) are based on a lower bound $m_i, i \in N$, for the number of deliverymen needed to serve a cluster. To tightly define these lower bounds, we solve a sequence of MIP models based on the SP defined for a vehicle route that goes from the depot to a cluster $i \in N$ and then back to the depot. For calculating $\underline{\eta}_i$, the SP is solved for every cluster $i \in N$ defining $l = M_L$ in constraints (3.40). Hereinafter, we shall refer to this problem as $SP_{cost}(i)$. For defining e_{il}, the SP is solved by changing its objective function to $w_{n_i+1} - w_{0_i}$, for every node $i \in N$ and deliverymen number $l \in L$. This problem will be denoted $SP_{time}(i, l)$. The value of m_i is assessed by the feasibility

of $SP_{time}(i,l)$. If this model is infeasible for a given l, then $m_i \geq l+1$. Otherwise, if it is feasible for every $l \in L$, then $m_i = 1$. When solving both $SP_{cost}(i)$ and $SP_{time}(i,l)$, we define a lower bound on the time spent in cluster $i \in N$ when served by $l \in L$ deliverymen as

$$\max\left\{\frac{1}{l}\sum_{h\in N^i} s_h, \max_{h\in N^i}\left\{t^i_{0_i h} + s_h + t^i_{h(n_i+1)}\right\}\right\}.$$

Notably, even though distances and travel times are proportional, the $SP_{cost}(i)$ and $SP_{time}(i,l)$ models yield different solutions due to the customers time windows and the possibility of serving the clusters with more than one deliveryman. This difference has already been explained and is illustrated in Figure 3.3 by showing a cluster i with four customers. Figure 3.3a presents the cluster data, indicating the cost of each arc and that each customer has a service time of 10 units (arcs cost, distance, and travel time are equivalent in the picture). If the limit on the number of deliverymen in each vehicle were $M_L = 2$, the solution to $SP_{cost}(i)$ would use only one of them to produce the second-level route portrayed in Figure 3.3b, since it is the shortest option with a total cost of 7 and total time of 47. Nevertheless, if the goal is to minimize the time spent in the cluster, using both deliverymen traveling the routes shown in Figure 3.3c would be the best choice, since the customers would be served in parallel, yielding a solution with total cost 10 and total time 25 for $SP_{time}(i,2)$. Hence, it is necessary to solve both $SP_{cost}(i)$ and $SP_{time}(i,l)$ for each node $i \in N$ and number of deliverymen $l \in L$. This is partly what creates the trade-off between deliveryman routes cost and time discussed in Section 3.3.



(a) Instance. (b) Cost minimization. (c) Time minimization.

Figure 3.3: Different solutions by minimizing deliveryman routes cost or time.

Calculating these lower bounds requires solving $n(M_L+1)$ MIP models: n times $SP_{cost}(i)$ and nM_L times $SP_{time}(i,l)$. Although computationally burdensome, this evaluation significantly improves the performance of the algorithms, as shown in Section 3.6.

3.5.4 Branch-and-Benders-cut

Given the exponential number of optimality and feasibility cuts, it is impractical to enumerate all of them a priori. Instead, the best approach is to solve the RMP and include optimality and feasibility cuts as needed in a BBC fashion (Moreno; Munari; Alem, 2019). To this extent, we solve the MP using a branch-and-cut algorithm that starts with the RMP. While processing the nodes of the branch-and-cut tree, every time a feasible integer solution to the RMP is found, we evaluate the corresponding SPs. If the solution of the RMP respects the optimality and feasibility cuts, we update the incumbent solution (if the new solution is better than the incumbent), otherwise we include the corresponding optimality and feasibility cuts and reoptimize the node.

The following steps represent the BBC algorithm:

1. Define cost and time lower bounds on the deliveryman routes in each cluster (Section 3.5.3);

2. Define the initial RMP and start the branch-and-cut method (Section 3.5.1);

3. Every time a feasible solution of the RMP is found in the branch-and-cut tree, check if the RCIs (3.34) are respected. If they are violated, include the corresponding RCIs and solve the current node again.

4. Separate the solution by vehicle routes, tighten the clusters time windows considering the vehicle route serving them, and solve the SPs (Section 3.5.2). For each SP, if it is feasible, include the corresponding optimality cuts (3.31), otherwise include the corresponding feasibility cuts (3.33). If all SPs are feasible and the solution cost updated with the deliveryman routes cost is lower than the incumbent cost, update the incumbent.

The algorithm terminates once all nodes of the branch-and-cut tree have been processed. In modern MIP solvers, this can be implemented using callbacks. To this extent, we declare the RMP model and start the solution procedure. In the callback, once an integer solution is found, the routine for separating RCIs (3.34), solving the SP, and separating optimality cuts (3.31) and feasibility cuts (3.33) is called (as described in the steps 3 and 4 above).

3.6 Computational experiments

We now describe the computational experiments performed to assess the performance of the proposed model and algorithms and their suitability to solve the VRPTWMD2R. The approaches were implemented in C++ and use Gurobi 10.0.2 with an optimality gap tolerance of 10^{-7}. The routines for separating RCIs (3.34) were implemented using the CVRPSep package (Lysgaard; Letchford; Eglese, 2004). The experiments were run on a computing cluster from Compute Canada, where each node is equipped with 2xAMD Rome 7532 processors running at 2.4GHz and up to 64GB of RAM for the CF, and 32GB for the BBC, with a time limit of 7,200s. For

each instance, we use 8 threads. For the MIP models $SP_{cost}(i)$ and $SP_{time}(i, l)$ that determine the lower bounds described in Section 3.5.3, we set a time limit of 10s; when the solver was unable to prove optimality within this time limit, we used the lower bound obtained by the solver to define the lower bounding parameter on time or cost. All instances and detailed results are available at `https://www.dep.ufscar.br/munari/vrptwmd/`.

Section 3.6.1 describes the instances used in our experiments. In Section 3.6.2, we present the results obtained with the CF and the different sets of VIs, allowing us to assess the effectiveness of the existing and new VIs. In Section 3.6.3, we discuss the results obtained with the BBC method. Finally, Section 3.6.4 provides managerial insights for this practical problem.

3.6.1 Instances

The generated instances are based on the Solomon (1987) instances for the VRPTW from classes C1, R1, and RC1. We considered that each node in a Solomon instance represents the parking location of a cluster in the VRPTWMD2R. Then, we generated one to seven customer locations around each parking location to create the customers in the corresponding cluster. Coordinates of the customers were generated following a normal distribution with mean in the parking location's coordinates and standard deviation $\sigma = 3$, which showed to be well suited for the problem representation. In the Solomon instances, only some nodes have time windows; if they do, i.e., the parking location has a time window, then time windows were generated for the customers assigned to them. These time windows were randomly generated considering the time window opening of the cluster and the average width of the clusters time windows, while ensuring feasibility of the instances. The service time of each customer is assumed to be the same as that of the corresponding parking locations.

We generated instances of five different sizes, namely 10–40, 15–60, 15–85, 20–80, and 25–125, in which the first number represents the number of clusters (parking locations) and the second number represents the total number of customers. This way, there are instances with 50, 75, 100, and 150 nodes, which are realistic for many last-mile logistics applications. There are 29 instances of each size, for a total of 145, all available online.

Following Pureza, Morabito, and Reimann (2012), we defined the cost parameters as $(f_v, c_v, f_d, c_d) = (1000, 10, 100, 1)$ and allowed up to $M_L = 3$ deliverymen per vehicle. The distances were calculated assuming Euclidean distances truncated to integers. For the vehicles, distance and travel time were considered equivalent, and deliverymen were assumed to travel at one-third of the vehicles' speed. After calculating distances and travel times, the Floyd-Warshall algorithm (Cormen et al., 2009) was run to ensure the triangular inequality was valid.

3.6.2 Compact formulation and valid inequalities

We first assess the performance of our CF (3.1)–(3.16), of the existing VIs (3.17)–(3.22), and of the newly proposed VIs (3.23)–(3.25). Table 3.1 shows the summarized results of the

experiments with the CF and VIs (detailed results are provided as supplementary material). It presents the results for the CF only (hereinafter referred to as CF1), the CF enhanced with VIs (3.17)–(3.22) from the literature (CF2), and the CF enhanced with VIs (3.18)–(3.25), both novel and literature-based (CF3). As discussed in Section 3.4, VIs (3.17) are redundant when VIs (3.25) are considered and are, therefore, not included in the latter scenario. In this table, "LR" stands for "LP relaxation", "LB" for "lower bound", "UB" for "upper bound", "Gap" for the optimality gap provided by the solver (as a percentage), "Time" for the running time in seconds, "# opt" for the number of instances for which the solver has proved optimality for the corresponding model, "# veh" for the number of vehicles in the best solution found, and "# del" for the number of deliverymen. All values represent the corresponding average, except for "# opt". We present the average gap as the average of optimality gaps of instances, not the gap calculated with the average LB and UB.

	Size	LR	LB	UB	Gap (%)	Time (s)	# opt	# veh	# del
CF1	10–40	1,027	5,508	7,138	26.93	6,069	5	3.69	8.45
	15–60	1,354	6,912	10,453	34.42	6,704	2	5.31	12.62
	15–85	1,398	6,900	12,403	46.47	6,954	1	6.34	15.69
	20–80	1,764	9,088	14,614	38.96	6,723	2	7.38	16.55
	25–125	2,071	11,446	20,426	45.63	6,954	1	10.38	23.83
	Total	1,523	7,971	13,007	38.48	6,681	11	6.62	15.43
CF2	10–40	4,063	5,727	7,170	21.51	6,081	7	3.72	8.38
	15–60	5,752	7,274	10,447	30.14	6,470	3	5.31	12.45
	15–85	6,023	7,379	12,255	40.50	6,954	1	6.24	15.66
	20–80	7,515	9,411	14,462	35.87	6,723	2	7.28	16.62
	25–125	10,066	12,017	20,220	41.62	6,953	1	10.24	23.79
	Total	6,684	8,362	12,911	33.93	6,636	14	6.56	15.38
CF3	10–40	4,637	7,131	7,131	0.00	316	28	3.69	8.41
	15–60	6,445	10,189	10,228	0.68	2,262	26	5.10	12.62
	15–85	7,757	12,018	12,116	1.26	4,108	13	6.14	15.69
	20–80	8,589	13,499	13,988	4.55	4,184	14	6.90	16.66
	25–125	12,653	18,358	19,195	5.76	5,155	9	9.48	24.24
	Total	8,016	12,239	12,532	2.45	3,205	90	6.26	15.52

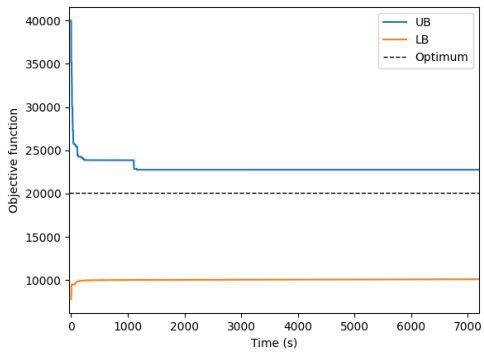
Table 3.1: Results of the experiments with CF and different sets of VIs.

These results indicate that the VIs significantly strengthen the LP relaxation of the CF. The inclusion of the VIs from the literature improves the average value of the LR in 338.92% and the novel VIs provide an additional improvement of 19.94%, leading to a total increase of 426.42% in the LR values. Moreover, for instances with sizes 15–85 and 25–125, the value of the LR of CF3 is higher than the final LB obtained after running the solver for two hours with the other two model configurations.

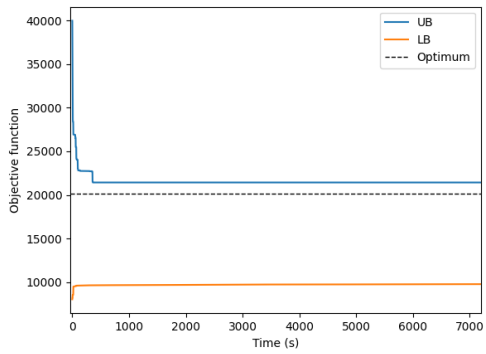
Regarding the performance of the MIP solver, the CF1 yields poor results, with high gaps even for the smallest instances. The solver proved optimality on few instances (11 out of a total of 145). The VIs from the literature (CF2) improve its performance, especially by lifting the

average LB in 4.91%, which yields a modest 4.55% improvement in the average gap. They also help prove optimality for three other instances, reaching 9.66% of the instances (14/145). Still, the average gap is 21.51% for the smallest-sized instances. Regarding the solution details, both the number of vehicles and the number of deliverymen used are reduced on average.

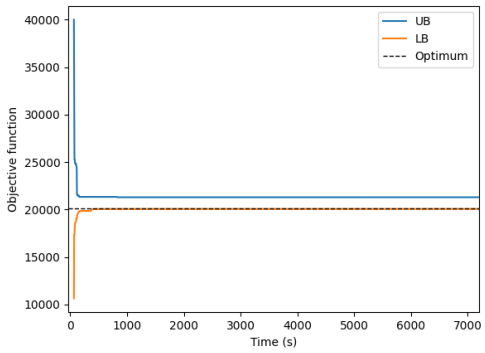
The combination of the VIs from the literature with the VIs proposed for the problem (CF3) produces a significant improvement in the results, leading to an additional 46.36% increase in the average LB and 31.48% reduction in the average gap. Furthermore, the number of instances with proven optimality increases to 90, which is more than half of the total instances, and more than six times the number of instances proved to optimality before. The runtime is significantly improved as a consequence of the new VIs and their effect in proving optimality. Note that the small instances can now be solved in about five minutes, and the average runtime is decreased by more than half. The comparison of the number of vehicles and deliverymen in the solutions of CF2 and CF3 shows that the number of vehicles is reduced while the number of deliverymen increases. This confirms the hypothesis presented in Section 3.3 that the deliverymen can be used to reduce the fleet size and, thus, the solution costs.



(a) CF1.



(b) CF2.



(c) CF3.

Figure 3.4: Convergence curves for instance R110 with size 25–125.

Figure 3.4 shows the convergence curves of the different CFs when solving instance R110 with size 25–125, which illustrates a common behavior of these models in many instances. Figures 3.4a and 3.4b indicate that both the CF1 and the CF2 start from high UBs that rapidly

decrease and the LB increases a little in the first few seconds. However, after 1000s of runtime, there is little improvement either in the UB or the LB, leading to large gaps (55.67% for the CF1 and 54.42% for the CF2). The CF3, as portrayed in Figure 3.4c, starts a few seconds later because it calculates the lower bounds discussed in Section 3.5.3 before starting the solution procedure. As in the other approaches, the UB rapidly decreases, but the difference here is the significant increase in the LB right in the first seconds of runtime. This figure illustrates the effect shown in Table 3.1. Indeed, although the improvement in the LR from using the novel VIs is small compared to the VIs from the literature, it significantly helps the performance of the MIP solver by increasing the LB throughout the branch-and-cut search tree. Nevertheless, these improvements do not overcome the tailing-off effect shown by the CF1 and the CF2, preventing the algorithm from proving optimality within the time limit, and finishing with an optimality gap of 5.80%. It is worth mentioning that, to assess whether longer runtimes would allow the solver to prove optimality for this instance, we have run the CF3 solving this specific instance with a time limit of twenty hours and, even though the gap was reduced, it was not possible to prove optimality.

These analyses have demonstrated the added value of the VIs from the literature and the significant improvement obtained with the newly proposed VIs for our problem. Using the CF3, the solver proved optimality for many instances and provided good bounds for the remaining larger instances. This version of the model is used in the next section to assess the performance of our BBC algorithm.

3.6.3 Branch-and-Benders-cut algorithm

Since the previous experiments clearly show the efficiency of the proposed VIs, in our BBC method the RMP always includes the VIs (3.20)–(3.24), and the SP includes VIs (3.17)–(3.19), (3.25), and (3.44).

The first experiments with the BBC method evaluate the relevance of the cut improvements discussed in Section 3.5.1, at first with constraints (3.5) and (3.14) in the MP instead of the RCIs (3.34). Table 3.2 presents the results considering two different versions of the method: BBC1 with route cuts (3.27) and (3.28); and BBC2 with improved path cuts (3.31) and (3.33). Notably, the performance of BBC2 is slightly worse for smaller instances but it is significantly better for larger ones. On average, the improved cuts yield positive impacts in the LB and UB, leading to a 0.19% gap improvement, 7.40% time reduction, and an additional instance proved to optimality. Given these results, the remaining experiments with the BBC are all run with the path cuts (3.31) and (3.33).

Table 3.3 compares the results of the experiments with CF3, BBC2, and BBC3 with improved cuts (3.31) and (3.33) and RCIs (3.34). Figure 3.5 shows the convergence curves of BBC2 for instance R110 with size 25–125. In addition to the LB and UB curves, this figure also shows the points in which optimality cuts were inserted.

	Size	LB	UB	Gap (%)	Time (s)	# opt	# veh	# del
BBC1	10–40	7,131	7,131	0.00	16	29	3.69	8.41
	15–60	10,200	10,228	0.49	631	28	5.10	12.62
	15–85	12,076	12,116	0.53	361	28	6.14	15.69
	20–80	13,398	13,972	4.99	1,948	22	6.90	16.66
	25–125	18,174	19,263	6.86	2,380	21	9.48	24.28
	Total	12196	12542	2.58	1067	128	6.26	15.53
BBC2	10–40	7,131	7,131	0.00	18	29	3.69	8.41
	15–60	10,196	10,228	0.56	435	28	5.10	12.62
	15–85	12,075	12,116	0.54	374	28	6.14	15.69
	20–80	13,425	13,972	4.89	2,003	22	6.90	16.62
	25–125	18,205	19,083	5.96	2,110	22	9.38	24.03
	Total	12,207	12,506	2.39	988	129	6.24	15.48

Table 3.2: Impact of cut improvements in the BBC method.

Compared to the CF3, the BBC2 algorithm reduces another 0.06% in the average gap and gives a slight improvement in the average UB for large instances of sizes 20–80 and 25–125. The greatest improvements, however, are in the number of instances solved to proven optimality and in the average runtime.

	Size	LB	UB	Gap (%)	Time (s)	# opt	# veh	# del
CF3	10–40	7,131	7,131	0.00	316	28	3.69	8.41
	15–60	10,189	10,228	0.68	2,262	26	5.10	12.62
	15–85	12,018	12,116	1.26	4,108	13	6.14	15.69
	20–80	13,499	13,988	4.55	4,184	14	6.90	16.66
	25–125	18,358	19,195	5.76	5,155	9	9.48	24.24
	Total	12,239	12,532	2.45	3,205	90	6.26	15.52
BBC2	10–40	7,131	7,131	0.00	18	29	3.69	8.41
	15–60	10,196	10,228	0.56	435	28	5.10	12.62
	15–85	12,075	12,116	0.54	374	28	6.14	15.69
	20–80	13,425	13,972	4.89	2,003	22	6.90	16.62
	25–125	18,205	19,083	5.96	2,110	22	9.38	24.03
	Total	12,207	12,506	2.39	988	129	6.24	15.48
BBC3	10–40	7,131	7,131	0.00	25	29	3.69	8.41
	15–60	10,196	10,228	0.56	448	28	5.10	12.55
	15–85	12,078	12,116	0.51	382	28	6.14	15.69
	20–80	13,430	13,975	5.06	2,010	22	6.90	16.62
	25–125	18,291	19,045	5.30	2,032	22	9.34	24.03
	Total	12,225	12,499	2.29	980	129	6.23	15.46

Table 3.3: Results of the experiments with the best versions of the CF and BBC approaches.

The BBC2 proved optimality for all instances with sizes 10–40, and for 28 out of 29 instances of sizes 15–60 and 15–85. In total, it proved optimality for 129 instances, which represents 88.97% of the total number of instances, and an increase of 43.33% compared to the CF3. Even for the instances with sizes 20–80 and 25–125, to which there was no improvement in the LB

and gap when comparing the BBC2 with the CF3, the number of instances solved to proven optimality went from 14 and 9 to 22 and 22 with the BBC2. For these sizes, the average LB and gap did not improve because the BBC2 performed worse than the CF3 in a few instances, despite being superior in most of them.

Moreover, the runtime was drastically reduced. Small instances were solved to optimality within seconds by the BBC2 method, and the average runtime, which was close to 2 hours for the CF1 and close to 1 hour for the CF3, was reduced to slightly more than 15 minutes. In part, this improvement is caused by the overcoming of the tailing-off effect, as shown in Figure 3.5. The BBC2 method proved optimality for that instance in less than 200s, while the other approaches had high gaps after 7200s and, as discussed in Section 3.6.2, could not prove optimality even after twenty hours of runtime. This leads to significant improvements in the runtime of the algorithm, with the average value representing 69.17% of reduction compared to the results of the CF3. In the instances of size 10–40, the runtime reduction is of 94.30%. This result is especially important considering that exact methods usually suffer from being very time-consuming, while the proposed BBC has presented reasonable running times for most instances.

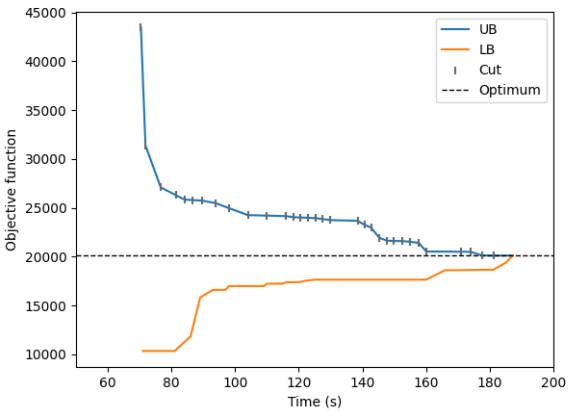


Figure 3.5: Convergence of the BBC2 method for instance R110 with size 25–125.

The inclusion of the RCIs (3.34) create LB and UB improvements that lead to an additional 0.10% gap improvement. The RCIs have more impact on instances of size 25–125, that show a 0.66% gap reduction. Considering the number of vehicles and deliverymen in the solutions, both from CF3 to BBC2 and from BBC2 to BBC3, there is a small reduction in both metrics.

Table 3.4 provides a closer look at the cuts inserted in the BBC2 algorithm. It displays the number of feasibility and optimality cuts inserted, as well as the total separation time for these cuts (which includes the time for solving the SPs). The number of feasibility cuts is less than one per instance on average. For the instances with size 15–60, no feasibility cut was needed in any instance. This indicates a very good performance of the proposed lower bounds for ensuring feasibility of the solution provided by the RMP. It also shows that many instances do not need feasibility cuts, as the one portrayed in Figure 3.5. The number of inserted optimality cuts grows with the instance sizes, but there are fewer than two cuts for each node on average. Figure 3.5

Size	# of feasibility	# of optimality	Total separation (s)
10–40	0.21	38.72	7.71
15–60	0.00	94.93	31.00
15–85	0.69	87.17	95.67
20–80	1.17	154.28	57.45
25–125	1.79	222.03	437.76
Avg	0.77	119.43	125.92

Table 3.4: Average number of cuts and separation times in the BBC algorithm.

illustrates the fact that when a new optimality cut is inserted, a new incumbent solution is often found, reducing the UB value.

Regarding the time spent separating these cuts, it grows rapidly with the instance sizes and is directly related to the number of cuts added. Additionally, each cut separation takes longer for larger instances as they have more customers in each route and larger clusters.

When comparing the results of our BBC3 approach with those obtained by solving the CF alone (CF1), the BBC yields a 36.19% reduction in the average gap, the number of instances solved to optimality is increased from 11 to 129, the average UB is improved by 3.91%, the average LB increases 53.37%, and the average runtime is reduced in 85.33%, which highlights the suitability of the proposed method to solve the problem.

Moreover, when looking at the results of the algorithms considering different instance sizes, it can be seen that it becomes more challenging to solve the problem as the instances grow. Nonetheless, different solution methods may be more or less sensitive to this increase in the difficulty in solving the problem depending if the size changes more expressively in the number of customers or clusters. Instances with sizes 15–85 and 20–80, for example, have a total of 100 nodes. On the one hand, CF1 and CF2 have better performances for instances with size 20–80 than for instances with size 15–85, indicating that the size of clusters affects these approaches more than the number of clusters. On the other hand, CF3 and BBC have better performances in the instances with size 15–85 than in those with size 20–80, suggesting that these methods are more affected by the number of clusters than by the cluster size. This shows that the proposed BBC method and the novel VIs were effective in decreasing the difficulty related to the second-level routes, as was our goal with those approaches given that these routes are less relevant (much cheaper and highly dependent on the other decisions) than the first-level routes and the number of vehicles or deliverymen used. Furthermore, the number of cuts added in the BBC for the instances with size 20–80 is 76.99% higher than for the instances with size 15–85, even though the separation time is 39.95% lower.

3.6.4 Managerial insights

We ran experiments to assess the relevance of considering deliveryman routes and to perform sensitivity analysis on the results. Experiments were run with the BBC method in a subset of

instances of sizes 15–85, 20–80, and 25–125 to which this method proved optimality in all configurations.

As discussed in Section 3.2, the previous works on the VRPTWMD ignored the deliveryman routes by considering that deliverymen have limited capacity and thus cannot visit more than one customer without returning to the vehicle. We adapted our methods to consider this alternative of having the deliverymen perform round trips to all customers in the cluster by simply setting the distance $d_{ij} = d_{i0} + d_{0j}$, where i and j are two customers and 0 represents the parking location. This new distance matrix effectively models the case of round trips to each customer. The travel times were defined accordingly. The results presented in Table 3.5 contrast this situation with the VRPTWMD2R proposed in this paper.

	Ignoring deliveryman routes	With deliveryman routes
# of vehicles	6.78	6.22
# of deliverymen	18.39	16.11
First-level distance	454.22	423.00
Second-level distance	669.00	474.56
Total cost	13,827.89	12,537.89

Table 3.5: The importance of considering deliveryman routes.

In spite of being a problem much easier to solve (the solution times were roughly one-third), ignoring the second-level routes creates significantly worse results. Since it overestimates the deliveryman routes time and distance, it has a greater need for both vehicles and deliverymen. The overall costs are 10.29% higher, highlighting the importance of considering the deliveryman routes in the problem.

These results also highlight that savings are expected if deliverymen can perform small routes instead of visiting one customer at a time. In applications where walking deliverymen cannot carry goods to serve more than one customer at a time, small scooters or cargo bikes can enable this. More generally, this analysis sheds light on the limitations and benefits of drone delivery, depending on the drone capacity and range.

Another important assessment is the trade-off between vehicle and deliveryman costs discussed in Section 3.3. Table 3.6 compares the results for three different cost structures, in which the first- and second-level cost components in (f_v, c_v, f_d, c_d) are set as follows: (*i*) deliverymen ten times cheaper than vehicles $(1000, 10, 100, 1)$; (*ii*) deliverymen and vehicles with the same costs $(100, 1, 100, 1)$; and (*iii*) deliverymen ten times more expensive than vehicles $(100, 1, 1000, 10)$. These results illustrate the trade-off mentioned above. They make clear that more efficient deliveryman routes and more deliverymen can be used to reduce both the number and the distance traveled by vehicles if this is interesting from a cost perspective. However, when this is not the case, the vehicles are more intensively used to reduce deliverymen costs. From the first scenario to the last, the average number of deliverymen per vehicle drops from 2.59 to 1.93, which is a 25.48% decrease. Nevertheless, the average distance traveled by each

vehicle and deliveryman does not change much from one scenario to the other since the fixed costs are much higher than the variable costs, enforcing that each vehicle and deliveryman is used as much as possible.

	Del $<$ Veh	Del $=$ Veh	Del $>$ Veh
# of vehicles	6.22	6.28	7.56
# of deliverymen	16.11	15.33	14.56
First-level distance	423.00	439.39	522.83
Second-level distance	474.56	467.22	438.17

Table 3.6: Costs sensitivity analysis.

Furthermore, we look at other possibilities of cost reduction enabled by clever uses of multiple deliverymen in practice. Table 3.7 presents a base case with a limit of $M_L = 3$ deliverymen in each vehicle in which they travel at one-third of the vehicles' speed. This base case is compared to another with a limit of $M_L = 5$ deliverymen that travel at the same speed. Once again, these results prove that deliverymen can be used to reduce the number of vehicles used. Here, the number of vehicles is reduced by 17.85% and the first-level distance is reduced accordingly by 12.54%. Despite the increase in deliverymen costs, this leads to an overall cost reduction of 11.32%. Another comparison is made with a case of fast deliverymen (twice the vehicles' speed), which could represent a case with drones, bicycles, or motorcycles as deliverymen, instead of walking carriers. Even though the number of deliverymen in each vehicle remains $M_L = 3$, this increased speed allows for a great reduction on the service time in each cluster, leading to better first-level routes. The average number of vehicles is reduced by 22.35%, the number of deliverymen by 25.88%, and the vehicles distance by 17.89%, leading to a cost reduction of 20.52%.

	Base case $M_L = 3$ 1/3 of vehicles' speed	More deliverymen $M_L = 5$ 1/3 of vehicles' speed	Fast deliverymen $M_L = 3$ 2$\times$ the vehicles' speed
# of vehicles	6.22	5.11	4.83
# of deliverymen	16.11	17.78	11.94
First-level distance	423.00	369.94	347.33
Second-level distance	474.56	530.33	464.06
Total cost	12,537.89	11,118.67	9,965.17

Table 3.7: Further advantages of multiple deliverymen.

A beneficial side effect of the business model incorporated by the VRPTWMD2R is a reduction on the emission of greenhouse gases (GHGs) and other pollutants. If the deliverymen are walking carriers, bicycles, or drones, for instance, GHGs emissions are much smaller in the second-level routes than in the first-level routes. Since the adoption of more deliverymen leads to reduced vehicle usage, as demonstrated above, this also reduces the environmental harm of the delivery. As presented in Table 3.7, the distance traveled by the vehicles can be reduced by

more than 15% with the proper usage of deliverymen, creating a much greener last-mile delivery system while reducing operational costs.

Finally, we assess the impact of the clustering decision in the VRPTWMD2R. Considering that clustering is part of the input data, there may be different instances with the same customers and parking locations, changing only the customer's clustering. We ran experiments to determine how this decision impacts the solution.

We analyzed, from our instances of size 10–40, which ones have customers that are not assigned to their closest parking locations with compatible time windows. This may happen due to the random generation of customers. There are 7 out of the 29 instances in which at least one customer is not assigned to its closest compatible parking location. Among those, in only four of them (C102, C103, C104, and RC104) there is an expressive number of these customers. All these instances share two characteristics: they have loose time windows and many parking locations with short distances among them.

We have then performed experiments to determine the impact of the clustering on the solution costs. To this extent, we generated customer locations as described in Section 3.6.1 for these four instances (C102, C103, C104, and RC104) of size 10–40 considering different values for the customers coordinates standard deviation: $\sigma = 3$ (original instances), $\sigma = 5$, and $\sigma = 7$. We have opted to evaluate instances with higher variability in the customer's positions since this is a factor that considerably affects clustering. We shall refer to these 12 instances (4 Solomon instances with customers generated based on 3 different standard deviation values) as instances with predefined clustering. For each instance with predefined clustering, we defined a new instance by reassigning each customer to the closest parking location with compatible time window (clustering based on proximity). The experiment consisted in comparing the solutions for the instances with predefined clustering against those with clustering based on proximity.

Table 3.8 presents the results of this evaluation, where "σ" indicates the customers' coordinates standard deviation, "Instance" is the instance name, and "Customers (%)" represents the percentage of customers that were reassigned to a closer cluster. Columns "Cost (%)", "Vehicles (%)", and "Deliverymen (%)" represent the variation in the solution cost, the number of vehicles, and the number of deliverymen, respectively, when comparing the optimal solution for the instance with predefined clustering and the instance with clustering based on proximity. A negative value indicates a reduction (better in the second clustering) and a positive value indicates an increase in the corresponding value.

The results of Table 3.8 for the original instances ($\sigma = 3$) show that, even for the four instances that were the most sensitive to clustering, the cost reduction that arises from the new clustering is small (less than 1% in average), with no changes in the number of vehicles or deliverymen used. This indicates that the clustering decision has little impact in the quality of the solution in situations in which customers are close to parking locations or have tight time windows, even if a considerable fraction of customers are not assigned to the closest possible parking location (more than 25% in the instances evaluated).

σ	Instance	Customers (%)	Cost (%)	Vehicles (%)	Deliverymen (%)
3 (original)	C102	30.00	-0.83	0.00	0.00
	C103	32.50	-0.69	0.00	0.00
	C104	40.00	-1.96	0.00	0.00
	RC104	25.00	-0.25	0.00	0.00
	Average	31.88	-0.93	0.00	0.00
5	C102	25.00	-1.01	0.00	0.00
	C103	25.00	-1.72	0.00	0.00
	C104	42.50	-35.55	-50.00	50.00
	RC104	30.00	-20.71	-20.00	-18.18
	Average	30.63	-14.75	-17.50	7.96
7	C102	42.50	-2.93	0.00	0.00
	C103	30.00	-1.19	0.00	0.00
	C104	47.50	-3.72	0.00	0.00
	RC104	42.50	-33.76	-37.50	-9.09
	Average	40.63	-10.40	-9.38	-2.27

Table 3.8: Solution variation after reclustering.

Nonetheless, when customers are more spread in a region with many candidate parking locations ($\sigma = 5$ and $\sigma = 7$), the clustering becomes more complicated and has more impact in the solution cost. Indeed, for some instances with a greater number of customers reassignments, the clustering has an impact on the vehicle fleet and deliverymen crew sizes, leading to cost reductions of more than 10% on average. It is worth noticing, however, that, even among these instances, the majority of them has a cost reduction of less than 5%. Also, these instances have loose time windows. In situations with tight time windows, the cost improvement would be much more restricted.

In conclusion, the presented results demonstrate the importance of properly evaluating the deliveryman routes and integrating them in a cost-effective manner with the vehicle routes. It has been shown that the adequate usage of deliverymen can reduce overall costs, the usage and number of vehicles, and the emission of GHGs and other pollutants. This creates the opportunity of devising less costly and greener operations. Regarding clustering, we have shown that it has little impact on the solutions of the VRPTWMD2R for instances with customers close to each other or with tight time windows.

3.7 Conclusion

In this work, we have introduced a novel problem in the literature called the vehicle routing problem with time windows, multiple deliverymen, and two-level routing. This problem is an extension of the vehicle routing problem with time windows and multiple deliverymen in which we incorporate the routes traveled by the deliverymen. We formally define and formulate the problem, propose valid inequalities for this formulation, and develop a branch-and-Benders-cut

algorithm to solve it efficiently.

The results of computational experiments show the relevance of including more than one deliveryman in each vehicle and properly optimizing their routes inside the clusters. We have shown that this evaluation leads to a significant cost reduction and directly impacts the number of vehicles and their routes in the solution. The experiments confirmed the suitability of the proposed methodology. The proposed BBC solves 129 out of 145 instances to proven optimality, with an average processing time of less than 1,000s. The proposed method is capable of solving instances of realistic sizes. Moreover, we have performed a sensitivity analysis on the costs that highlighted opportunities to improve the usage of multiple deliverymen, such as increasing the number of deliverymen in each vehicle and adopting faster deliverymen (e.g., drones, bicycles, and motorcycles). We have also discussed beneficial environmental effects of this business model, which are relevant in urban logistics.

Finally, some possibilities of future work are extending the problem further and proposing other solution methods. Interesting extensions would be considering pickup-and-delivery schemes, evaluating heterogeneous fleet (especially in the first level), dealing with uncertain data (e.g., uncertainties in the demand or travel times), or integrating the clustering decision in the optimization. Regarding new methods, the development of heuristics and metaheuristics, or their combination with the proposed BBC to create hybrid methods, could lead to good solutions for even larger instances.

4 Last-mile delivery with multiple deliverymen: formulation and exact solution methods for a rich vehicle routing problem

Abstract

There is an increasing demand for cost- and time-efficient last-mile delivery due to the growth of urban areas and the expansion of home-delivery systems. To respond to this need, many companies and academics have focused on proposing inventive delivery schemes to reduce costs and improve the service level offered to customers, while dodging traffic and avoiding an increase in the emission of green house gases and other pollutants, such as combining the use of vehicles with walking carriers. The vehicle routing problem with time windows and multiple deliverymen is an example of such delivery systems. In this problem, each vehicle may travel with more than one deliveryman to serve many customers with a single stop of the vehicle and reduce the overall time that the vehicle stays parked. As originally defined, this problem considers that the definition of which customers are served from each parking location and the routes traveled by the deliveryman can be predefined. We propose a variant of this problem that includes both of these decisions in the optimization. The novel problem is formally defined and formulated. Theoretical properties and useful lower bounds are introduced and used to propose several valid inequalities. The problem is also decomposed in a Benders scheme and solved exactly by a branch-and-Benders-cut algorithm. Extensive computational experiments show the suitability of the proposed methodology to solve the problem. Furthermore, managerial insights indicate that the inclusion of the customer clustering and deliveryman routes in the optimization leads to an average cost reduction of over 10%, with this value being much higher for particular instances.[1]

[1] This chapter is a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos).

4.1 Introduction

Last-mile delivery is a growing concern in logistics operations due to the increasing demand for efficient deliveries in cities caused by the urban population growth and the expansion of e-commerce (Bayliss et al., 2023). Compared to traditional routing problems, last-mile delivery systems encompass additional challenges such as finding places to park the vehicles and poor traffic conditions (Martinez-Sykora et al., 2020). Also, some cities have restrictions on vehicle sizes and circulation.

On the one hand, the aforementioned issues are faced by the logistics companies when designing their routes and delivery systems. On the other hand, from the public perspective, poorly designed delivery systems negatively affect the traffic and can lead to higher emission of greenhouse gases (GHGs) and other pollutants (Bektaş; Laporte, 2011). These questions show the importance of evaluating and designing more effective last-mile delivery systems.

A common approach in these systems is to use two-echelon schemes (Cuda; Guastaroba; Speranza, 2015; Sluijk et al., 2023), in which larger vehicles take the goods from the depot to transshipment facilities (satellites) and smaller vehicles take them from these facilities to the final customers. The main examples are the two-echelon vehicle routing problem (2E-VRP) and the two-echelon location routing problem (2E-LRP).

Another possibility is to use two-echelon schemes without having these transshipment satellites by using the first-echelon vehicles as mobile facilities and having smaller vehicles taking goods from the vehicles to the customers. Some common applications include having the customers served by drones (Moshref-Javadi; Winkenbach, 2021), robots (Alfandari; Ljubić; De Melo da Silva, 2022), or carriers on bicycles or walking (Cabrera; Cordeau; Mendoza, 2022; Bayliss et al., 2023; Senna et al., 2024a). These smaller vehicles take the goods directly from the vehicles to the customers, with no need for transshipment facilities while increasing the efficiency of the deliveries. This is highly beneficial since it does not incur in additional costs of facility location (2E-LRP) and transshipment (2E-VRP), and does not require infrastructure investments in satellites. Additionally, the use of greener options in the second echelon (drones, robots, and people walking or cycling) leads to reducing the emission of GHGs and pollutants and does not impact the traffic.

A particularly interesting application is the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD). This problem is based on a depot, a set of customers, and a set of potential parking locations. Vehicles take goods from the depot to the customers and, once parked, the deliverymen traveling with this vehicle serve the customers. It is assumed that each vehicle may carry more than one deliveryman and that they serve the customers in parallel, reducing the time that the vehicle stays parked throughout the route. Since vehicle costs are often higher than deliveryman costs, this creates an opportunity for cost reduction (Pureza; Morabito; Reimann, 2012).

The common approach in the VRPTWMD is to assume that the deliveryman routes and the

definition of which customers are to be served by each parking location (clustering) can be pre-processed (Pureza; Morabito; Reimann, 2012; Álvarez; Munari, 2017; Munari; Morabito, 2018; De La Vega; Munari; Morabito, 2020). The customer clustering has been addressed by Senarclens de Grancy and Reimann (2015) and Senarclens de Grancy (2015), and the deliveryman routes by Senna et al. (2024a), all of them showing the benefits of including these decisions on the problem. However, to the best of our knowledge, no other work has evaluated the impact of including both the deliveryman routes and the customer clustering in the optimization problem. In this paper, we extend the VRPTWMD by considering both of these decisions. The contributions of this paper are sixfold:

- The introduction of a variant of the VRPTWMD that encompasses the decisions on which customers are to be served by a vehicle parked at each parking location and the deliveryman routes;
- The proposition of a mixed-integer programming formulation to represent this problem;
- The discussion of theoretical properties of the problem and the proposition of valid inequalities;
- A branch-and-Benders-cut algorithm to solve the problem based on Benders decomposition;
- Extensive computational experiments evaluating the performance of the proposed solution approaches;
- Managerial insights that highlight the importance of including these decisions in the problem.

The remaining of this paper is structured as follows. In Section 4.2, a brief literature review of the VRPTWMD is presented. Section 4.3 defines the problem. In Section 4.4, the problem is formulated, some properties are discussed, and valid inequalities are proposed. In Section 4.5, a Benders decomposition is proposed along with a branch-and-Benders-cut algorithm. Section 4.6 presents the computational experiments and provides some interesting managerial insights. Section 4.7 discusses concluding remarks.

4.2 Literature review

The VRPTWMD was proposed by Pureza, Morabito, and Reimann (2012) to reflect the deliveries of a beverage company in a densely populated urban center. In large cities, it is common that vehicles spend long times in their deliveries traveling slowly (due to traffic) in search of a place to park. It is also common that there are many customers close to each other, creating an opportunity of serving such customers with a single stop of the vehicle. Although reducing the

issues of traffic and lack of parking location availability, this approach has the downside of having the vehicles parked during long periods while a single deliveryman serves many customers. To speed up the delivery process, this company came up with the idea of including more than one deliveryman in each vehicle, reducing the time that the vehicles stay parked and increasing the delivery efficiency. Since the costs associated with vehicles are usually higher than those of deliverymen, this creates an opportunity for cost reduction. Moreover, since deliverymen emit less GHGs and other pollutants, this business model has the beneficial side effect of reducing emissions.

Based on the operations of the beverage company studied, (Pureza; Morabito; Reimann, 2012) defined the problem with two simplifying hypotheses: (*i*) the definition of the customers to be served by each vehicle stop (clusters) is predefined, and (*ii*) the deliverymen routes inside each cluster can be defined in a preprocessing phase.

Since then, most works that studied the VRPTWMD followed these ideas. Senarclens de Grancy and Reimann (2014) and Álvarez and Munari (2016) compared the performance of different metaheuristics to solve the problem. Souza Neto and Pureza (2016) extended the problem to include the possibility of multiple trips for the vehicles. Munari and Morabito (2018) proposed the first exact algorithm for the VRPTWMD: a branch-and-price, solving the problem with column generation. Álvarez and Munari (2017) combined this method with two metaheuristics to create a hybrid algorithm. De La Vega, Munari and Morabito looked at the problem with uncertainties by means of robust optimization heuristically (De La Vega; Munari; Morabito, 2019) and exactly (De La Vega; Munari; Morabito, 2020).

These hypotheses make sense in beverage delivery schemes, since the goods to be transported are usually large and heavy. Thus, a walking deliveryman cannot travel far from the vehicle while transporting these commodities, and the clusters can be easily defined based on customers that have compatible time windows and are very close to each other. Furthermore, the walking deliveryman would not be capable of serving many customers without coming back to the vehicle to collect more goods before heading to the next customer. This way, the deliveryman routes become trivial as back and forth trips from the vehicle to the customers.

Nonetheless, in different applications that include smaller demands or larger deliveryman capacities, the definition of customer clusters and deliveryman routes are not so straightforward and their inclusion in the optimization problem becomes beneficial. Senarclens de Grancy and Reimann (2015) were the first to realize this and propose the inclusion of the clustering in the problem. In fact, they proposed two novel heuristics to define the clusters. Senarclens de Grancy (2015) went further to combine the clustering with the routing. Both of these works looked at the VRPTWMD by removing the hypothesis (*i*) of predefined clustering while maintaining the hypothesis (*ii*) that the deliveryman routes should be predefined. Senna et al. (2024a) looked at the problem from a different perspective, by evaluating the deliveryman routes and, hence, removing hypothesis (*ii*) that they should be preprocessed. However, they still considered that the clusters would be defined in a preprocessing phase as stated by hypothesis (*i*). These three works

proved the relevance of extending the VRPTWMD in these ways and the benefits it creates. Nevertheless, there is no work that addressed removing both of these simplifying hypotheses to include the customer clustering and the deliveryman routes in the optimization problem.

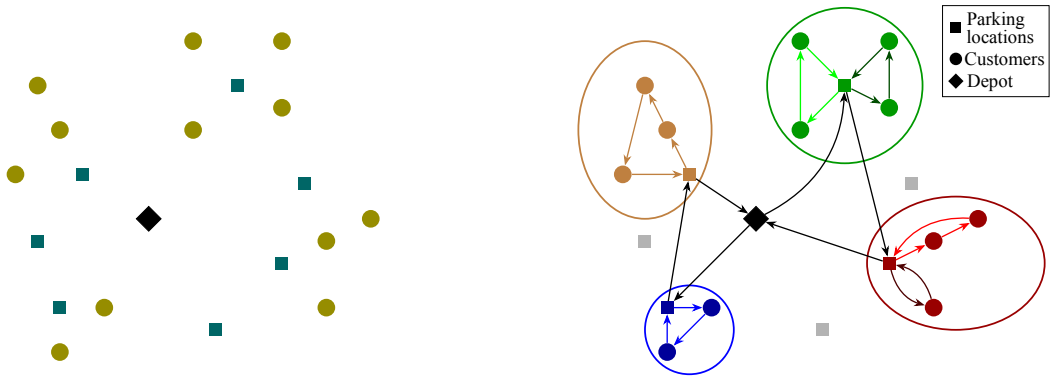
In this paper, we aim at bridging this gap, by defining the vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing (VRPTWMDC2R). The VRPTWMDC2R extends the VRPTWMD by including both the customer clustering and the deliverymen routes in the optimization. As discussed above, this is especially interesting when having deliverymen with large capacities compared to customer demands. Moreover, when considering time windows, this becomes even more important. Upon preprocessing the clusters and/or deliveryman routes, the order of visits inside a cluster must be predefined and, hence, there is little flexibility regarding time of arrival at each parking location, since one must ensure that the vehicle would arrive with time to have the deliverymen serving the customers within their time windows in the predetermined order. As discussed by Senna et al. (2024a), this would require a preprocessing in the time windows of the parking locations that could make the problem more constrained than it actually is, possibly leading to worse solutions.

4.3 Problem definition

The VRPTWMDC2R is defined over a directed graph $G = (N, A)$, with N representing the set of nodes and A the set of arcs. Let N^1 be the set of the n potential parking locations and N^2 the set of customers. We represent the depot by 0 (source) and $n + 1$ (sink) and extend the set N^1 by defining $N_0^1 = N^1 \cup \{0, n + 1\}$. The set of nodes is defined as $N = N_0^1 \cup N^2$. The set $A^1 = \{(i, j) : i, j \in N_0^1, i \neq j, i \neq n + 1, j \neq 0\}$ encompasses every arc that connects two parking locations or the depot and a parking location. The set $\widetilde{A}^2 = \{(i, j) : i, j \in N^2, i \neq j\}$ contains the arcs that connect every pair of customers. Let $(N^1 : N^2)$ represent the arcs that go from a node in N^1 to a node in N^2. We shall denote by $A^2 = \widetilde{A}^2 \cup (N^1 : N^2) \cup (N^2 : N^1)$ the set of arcs connecting two customers or a customer and a parking location. The set of arcs in the graph is denoted by $A = A^1 \cup A^2$.

A homogeneous fleet of vehicles with limits of capacity Q^1 and route duration T travels in the arcs of A^1. Each vehicle may carry from 1 to M_L deliverymen that will serve the customers. Once the vehicle is parked, the deliverymen leave the vehicle to serve the customers. We assume that the deliverymen travel with the same vehicle throughout the whole vehicle route, i.e., the vehicle is parked while the deliverymen serve the customers and it waits all of them to come back before traveling to the next cluster. Also, each deliverymen may perform at most one route per vehicle stop and has a capacity Q^2. The deliverymen travel in the arcs of set A^2. We assume that each parking location has a limited transshipment capacity to reflect the fact that it is not viable to serve an indefinite amount of demand from a single parking location.

Figure 4.1 presents an example of the VRPTWMDC2R. Figure 4.1a illustrates an instance of the problem, with a depot, a set of customers, and a set of potential parking locations. In



(a) Instance. (b) Solution.

Figure 4.1: An illustrative example of the VRPTWMDC2R.

Figure 4.1b a feasible solution is portrayed. Only four out of the seven potential parking locations are effectively used, and the customers are clustered around these locations. The black arrows represent the vehicle routes, and the colorful arrows inside the clusters correspond to the deliveryman routes. The vehicle on the right-hand side of the figure travels with two deliverymen. Once it arrives at the upper right green cluster, the deliverymen leave the vehicle to serve the customers in parallel. Afterwards, they return to the vehicle and travel to the lower right red cluster, when the same procedure is repeated. Then, the vehicle returns to the depot. The vehicle on the left-hand side of the picture travels with a single deliveryman, who serves all customers in the clusters visited by this vehicle.

Every cluster is visited by exactly one vehicle and every customer by exactly one deliveryman. Both customers and parking locations have time windows for the vehicle or deliveryman arrival, but there is no time limit for the departure from the node. Every customer has a positive demand that must be completely fulfilled. We assume that both the vehicle fleet and the deliveryman crew are unlimited. The decisions of the problem are (*i*) which customers are to be assigned to each parking location, (*ii*) the number of vehicles to be used, (*iii*) the vehicle routes, (*iv*) the number of deliverymen traveling with each vehicle, and (*v*) the deliveryman routes.

4.4 Mathematical formulation

To improve understanding, variables and parameters associated with customers and deliverymen routes are identified with a superscript "2" (second echelon) and the ones associated with vehicles and parking locations with a superscript "1" (first echelon). Also, nodes in N_0^1 are represented by i and j, and nodes in N^2 by h and k. We define $L = \{1, 2, \ldots, M_L\}$ as the set of possible configurations (number) of deliverymen on a vehicle. The parameters included in the formulation are:

M_L Maximum number of deliverymen in each vehicle;

f^1 Fixed cost associated with each vehicle;

c^1 Unitary distance cost of vehicle routes;

f^2 Fixed cost associated with each deliveryman;

c^2 Unitary distance cost of deliveryman routes;

Q^1 Vehicle load capacity;

Q^2 Deliveryman load capacity;

d^1_{ij} Distance between nodes i and $j, (i, j) \in A^1$ (asymmetrical);

t^1_{ij} Travel time between nodes i and $j, (i, j) \in A^1$ (asymmetrical);

H_i Capacity of parking location $i \in N^1$;

s^1_i Lower bound for the time spent in parking location $i \in N^1$ if this parking location is used. Defined as $s^1_i = \min_{h \in N^2}\{t^2_{ih} + s^2_h + t^2_{hi}\}, \forall\, i \in N^1$;

$[a^1_i, b^1_i]$ Parking location $i \in N^1$ time window. We denote as $T = b^1_{n+1}$ the travel time limit of each vehicle;

d^2_{hk} Distance between nodes h and $k, (h, k) \in A^2$ (asymmetrical);

t^2_{hk} Travel time between nodes h and $k, (h, k) \in A^2$ (asymmetrical);

q^2_h Demand of customer $h \in N^2$;

s^2_h Service time of customer $h \in N^2$;

$[a^2_h, b^2_h]$ Time window of customer $h \in N^2$.

The variables of the problem are:

x^1_{ijl} Binary variable that indicates whether a vehicle travels from node i to node j with l deliverymen, $(i, j) \in A^1, l \in L$;

w^1_i Arrival time at node $i \in N^1$;

w'^1_i Departure time from node $i \in N^1$;

u^1_i Vehicle load after leaving node $i \in N^1$;

x^2_{hk} Binary variable that indicates whether a deliveryman travels through arc $(h, k) \in A^2$;

w^2_h Instant in which service at customer $h \in N^2$ begins;

u^2_h Deliveryman load after leaving customer $h \in N^2$;

z_{jh} Binary variable that indicates whether customer $h \in N^2$ is served by a deliveryman traveling with a vehicle parked at parking location $j \in N^1$.

We present the following compact formulation (CF) to formally define the VRPTWMDC2R:

$$\text{(CF) min} \sum_{j\in N^1}\sum_{l\in L}(f^1 + lf^2)x^1_{0jl} + c^1 \sum_{(i,j)\in A^1}\sum_{l\in L} d^1_{ij}x^1_{ijl} + c^2 \sum_{(h,k)\in A^2} d^2_{hk}x^2_{hk} \tag{4.1}$$

$$\text{s.t.} \sum_{i:(i,j)\in A^1}\sum_{l\in L} x^1_{ijl} \leq 1,\ \forall\, j \in N^1 \tag{4.2}$$

$$\sum_{i:(i,j)\in A^1} x^1_{ijl} = \sum_{i:(j,i)\in A^1} x^1_{jil},\ \forall\, j \in N^1, l \in L \tag{4.3}$$

$$\sum_{i\in N^1} x^1_{0il} = \sum_{i\in N^1} x^1_{i(n+1)l},\ \forall\, l \in L \tag{4.4}$$

$$\sum_{h\in N^2} q^2_h z_{ih} \leq H_i,\ \forall\, i \in N^1 \tag{4.5}$$

$$\sum_{j\in N^1} z_{jh} = 1,\ \forall\, h \in N^2 \tag{4.6}$$

$$\sum_{h:(h,k)\in A^2} x^2_{hk} = 1,\ \forall\, k \in N^2 \tag{4.7}$$

$$\sum_{h:(h,k)\in A^2} x^2_{hk} = \sum_{h:(k,h)\in A^2} x^2_{kh},\ \forall\, k \in N^2 \tag{4.8}$$

$$\sum_{h\in N^2} x^2_{ih} = \sum_{h\in N^2} x^2_{hi},\ \forall\, i \in N^1 \tag{4.9}$$

$$\sum_{k\in N^2} x^2_{jk} \leq \sum_{i:(i,j)\in A^1}\sum_{l\in L} l x^1_{ijl},\ \forall\, j \in N^1 \tag{4.10}$$

$$x^2_{hk} + x^2_{kh} + z_{ih} - z_{ik} \leq 1,\ \forall\, h,k \in N^2, h \neq k, i \in N^1 \tag{4.11}$$

$$x^2_{ih} \leq z_{ih},\ \forall\, (i,h) \in (N^1 : N^2) \tag{4.12}$$

$$x^2_{hi} \leq z_{ih},\ \forall\, (h,i) \in (N^2 : N^1) \tag{4.13}$$

$$w'^1_i \geq w^1_i,\ \forall\, i \in N^1 \tag{4.14}$$

$$w^1_j \geq w'^1_i + t^1_{ij} - M_{ij}\left(1 - \sum_{l\in L} x^1_{ijl}\right),\ \forall\, i,j \in N^1, i \neq j \tag{4.15}$$

$$w^2_k \geq w^2_h + s^2_h + t^2_{hk} - M_{hk}(1 - x^2_{hk}),\ \forall\, (h,k) \in \widetilde{A}^2 \tag{4.16}$$

$$w^2_k \geq w^1_i + t^2_{ik} - M_{ik}(1 - x^2_{ik}),\ \forall\, (i,k) \in (N^1 : N^2) \tag{4.17}$$

$$w'^1_i \geq w^2_h + s^2_h + t^2_{hi} - M_{hi}(1 - x^2_{hi}),\ \forall\, (h,i) \in (N^2 : N^1) \tag{4.18}$$

$$u^1_i \geq \sum_{h\in N^2} q^2_h z_{ih},\ \forall\, i \in N^1 \tag{4.19}$$

$$u^1_j \geq u^1_i + \sum_{h\in N^2} q^2_h z_{jh} - Q^1\left(1 - \sum_{l\in L} x^1_{ijl}\right),\ \forall\, i,j \in N^1, i \neq j \tag{4.20}$$

$$u^2_k \geq u^2_h + q^2_k - Q^2(1 - x^2_{hk}),\ \forall\, (h,k) \in \widetilde{A}^2 \tag{4.21}$$

$$x^1_{ijl} \in \{0,1\},\ \forall\,(i,j) \in A^1, l \in L \tag{4.22}$$
$$0 \leq u^1_i \leq Q^1,\ \forall\, i \in N^1 \tag{4.23}$$
$$a^1_i \leq w^1_i \leq b^1_i,\ \forall\, i \in N^1 \tag{4.24}$$
$$a^1_i + s^1_i \leq w'^1_i \leq T - t^1_{i(n+1)},\ \forall\, i \in N^1 \tag{4.25}$$
$$x^2_{hk} \in \{0,1\},\ \forall\,(h,k) \in A^2 \tag{4.26}$$
$$a^2_h \leq w^2_h \leq b^2_h,\ \forall\, h \in N^2 \tag{4.27}$$
$$q^2_h \leq u^2_h \leq Q^2,\ \forall\, h \in N^2 \tag{4.28}$$
$$z_{jh} \in \{0,1\},\ \forall\, j \in N^1, h \in N^2. \tag{4.29}$$

The objective function (4.1) minimizes fixed and variable costs of both vehicles and deliverymen. Constraints (4.2) limit the usage of each parking location to at most once. Constraints (4.3) and (4.4) are vehicle flow conservation. Constraints (4.5) limit the demand served by each parking location to its load capacity. Constraints (4.6) ensure that each customer is assigned to exactly one parking location and constraints (4.7) that every customer is visited exactly once. Constraints (4.8) and (4.9) are the deliveryman routes equivalent to (4.3) and (4.4). Constraints (4.10) limit the number of deliverymen leaving a parking location to serve the customers to the number of deliverymen that arrive at that parking location. Constraints (4.11) ensure that deliverymen can only travel between nodes assigned to the same parking location. They are adapted from the formulation that was proposed to the 2E-LRP by Senna et al. (2024b). Constraints (4.12) and (4.13) define that a deliveryman can only travel between a parking location and a customer if this customer has been assigned to that parking location. Constraints (4.14) state that a vehicle can only leave a parking location after arriving at it. Constraints (4.15) and (4.16) define the time flow in vehicle and deliveryman routes, respectively. Constraints (4.17) and (4.18) synchronize vehicle and deliveryman routes. In these constraints, $M_{ij} = \max\{0, T - t^1_{i(n+1)} + t^1_{ij} - a^1_j\}$, $M_{hk} = \max\{0, b^2_h + s^2_h + t^2_{hk} - a^2_k\}$, $M_{ik} = \max\{0, b^1_i + t^2_{ik} - a^2_k\}$, and $M_{hi} = \max\{0, b^2_h + s^2_h + t^2_{hi} - a^1_i - s^1_i\}$. Constraints (4.19) define that the load of a vehicle after visiting a cluster is at least the sum of the demands assigned to the corresponding parking location. Constraints (4.20) and (4.21) control the load flow of vehicle and deliveryman routes, respectively. Constraints (4.22)–(4.29) define the variable domains.

4.4.1 Theoretical properties

In this section we present some theoretical properties of the problem and establish useful lower bounds that are used to define our valid inequalities and solution methods. These results are formally defined and proved in Propositions 4.1 to 4.5 and Corollary 4.1.

Proposition 4.1. *If the triangular inequality holds, there is an optimal solution in which only parking locations with customers assigned to it are visited.*

Proof. Suppose that a vehicle visits nodes $i, j, k \in N^1$ in this sequence and that there is no

customer assigned to parking location j. In this case, if the vehicle goes straight from node i to node k, the route is still feasible and the total distance is reduced by $d^1_{ij} + d^1_{jk} - d^1_{ik}$. Given that the triangular inequality holds, $d^1_{ij} + d^1_{jk} \geq d^1_{ik}$ and, hence, $d^1_{ij} + d^1_{jk} - d^1_{ik} \geq 0$, this "shortcut" leads to a vehicle route that is at most as costly as the previous one. Therefore, given a solution that visits a parking location without customers assigned to it, there is always a solution that is at least as good as this one and does not visit this parking location. □

Corollary 4.1. *If the triangular inequality holds, there is an optimal solution in which a deliveryman leaves every parking location visited by a vehicle.*

Proposition 4.2. *If the triangular inequality holds, a lower bound on the time spent in a parking location $i \in N^1$ visited by l deliverymen is given by*

$$\sum_{h \in N^2} \left(s^2_h + (t^2_{ih} + t^2_{hi}) \frac{q^2_h}{Q^2} \right) \frac{z_{ih}}{l}.$$

Proof. Consider an instance of the asymmetric capacitated vehicle routing problem (ACVRP) with the depot represented by 0, the set of customers by N, the demands of a node $j \in N$ by q_j, and the travel time to and from the depot as t_{j0} and t_{0j}, respectively. Given this notation, one can show that the total travel time of the vehicles (summing up for all vehicles) in this instance is at least $\sum_{j \in N}(t_{0j} + t_{j0})\frac{q_j}{Q}$ by extending to the ACVRP the lower bound presented by Haimovich and Kan (1985) for the capacitated vehicle routing problem (CVRP) – following the logic presented in their paper.

In our problem, once defined the customers assigned to a parking location, the dynamics of the deliverymen inside this cluster are similar to a vehicle routing problem with time windows (VRPTW) in which the parking location acts as the depot. Since the VRPTW is a more constrained version of the ACVRP, $\sum_{h \in N^2}(t^2_{ih} + t^2_{hi})\frac{q^2_h}{Q^2} z_{ih}$ is a lower bound on the total travel time inside cluster $i \in N^1$. The time spent in the cluster considers both the total travel time and the total service time. Also, if the cluster is visited by l deliverymen, in a best case scenario the total time is evenly divided between these deliverymen, yielding the lower bound presented above. □

Proposition 4.3. *If the triangular inequality holds, a lower bound on the total travel time of the vehicles is*

$$\frac{1}{Q^1} \sum_{i \in N^1} \sum_{h \in N^2} (t^1_{0i} + t^1_{i0}) q^2_h z_{ih}.$$

Proof. Analogous to the proof of Proposition 4.2. □

Proposition 4.4. *If the triangular inequality holds, a lower bound on the total time the vehicles stay out of the depot is*

$$\frac{1}{Q^1} \sum_{i \in N^1} \sum_{h \in N^2} (t^1_{0i} + t^1_{i0}) q^2_h z_{ih} + \frac{1}{M_L} \sum_{h \in N^2} \left(s_h + \frac{1}{Q^2} \sum_{i \in N^1} (t^2_{ih} + t^2_{hi}) q^2_h z_{ih} \right).$$

Proof. By summing up the lower bound from Proposition 4.2 for all parking locations considering that they are visited by M_L deliverymen (resulting in the smallest possible lower bound) with the lower bound from Proposition 4.3 for the total travel time of the vehicles, one gets this lower bound. □

Proposition 4.5. *If the triangular inequality holds, a lower bound on the cost of the deliverymen routes inside a cluster $i \in N^1$ is*

$$c^2 \sum_{h \in N^2} (d^2_{ih} + d^2_{hi}) q^2_h z_{ih}.$$

Proof. Analogous to the proof of Proposition 4.2. □

4.4.2 Valid inequalities

With these results, the presented CF can be strengthened by the following valid inequalities (VIs):

$$\sum_{(i,j) \in A^1 : i,j \in S} \sum_{l \in L} x^1_{ijl} \leq |S| - 1, \ \forall \, S \subset N^1 : |S| \in \{2,3\} \tag{4.30}$$

$$\sum_{(h,k) \in A^2 : h,k \in S} x^2_{hk} \leq |S| - 1, \ \forall \, S \subset N^2 : |S| \in \{2,3\} \tag{4.31}$$

$$\sum_{j \in N^1} \sum_{l \in L} x^1_{0jl} \geq \left\lceil \frac{1}{Q^1} \sum_{h \in N^2} q^2_h \right\rceil \tag{4.32}$$

$$\sum_{\substack{(i,j) \in A^1 \\ i \neq 0}} \sum_{l \in L} l x^1_{ijl} \geq \left\lceil \frac{1}{Q^2} \sum_{h \in N^2} q^2_h \right\rceil \tag{4.33}$$

$$x^1_{ijl} = 0, \ \forall \, i,j \in N^1, i \neq j, l \in L : a^1_i + s^1_i + t^1_{ij} > b^1_j \tag{4.34}$$

$$x^2_{hi} = 0, x^2_{ih} = 0, z_{ih} = 0, \ \forall \, i \in N^1, h \in N^2 : \tag{4.35}$$
$$(a^1_i + t^2_{ih} > b^2_h) \vee (a^2_h + s^2_h + t^2_{hi} > T - t^1_{i(n+1)})$$

$$x^2_{hk} = 0, \ \forall \, h,k \in N^2, h \neq k : (a^2_h + s^2_h + t^2_{hk} > b^2_k) \vee (q^2_h + q^2_k > Q^2) \tag{4.36}$$

$$\sum_{\substack{(i,j) \in A^1 \\ i \neq 0}} \sum_{l \in L} x^1_{ijl} \geq P_{min} \tag{4.37}$$

$$x^2_{jk} \leq \sum_{i : (i,j) \in A^1} \sum_{l \in L} x^1_{ijl}, \ \forall \, j \in N^1, k \in N^2 \tag{4.38}$$

$$\sum_{k \in N^2} x^2_{jk} \geq \sum_{i : (i,j) \in A^1} \sum_{l \in L} x^1_{ijl}, \ \forall \, j \in N^1 \tag{4.39}$$

$$\sum_{h \in N^2} z_{jh} \geq \sum_{i : (i,j) \in A^1} \sum_{l \in L} x^1_{ijl}, \ \forall \, j \in N^1 \tag{4.40}$$

$$z_{jh} \leq \sum_{i : (i,j) \in A^1} \sum_{l \in L} x^1_{ijl}, \ \forall \, j \in N^1, h \in N^2 \tag{4.41}$$

$$\sum_{h \in N^2} z_{jh} \geq \sum_{h \in N^2} x^2_{jh}, \ \forall \, j \in N^1 \tag{4.42}$$

$$z_{jh} \leq \sum_{k \in N^2} x^2_{jk}, \ \forall \, j \in N^1, h \in N^2 \tag{4.43}$$

$$\sum_{h \in N^2} q^2_h z_{jh} \leq Q^2 \sum_{i:(i,j) \in A^1} \sum_{l \in L} l x^1_{ijl}, \ \forall \, j \in N^1 \tag{4.44}$$

$$w'^1_i \geq a^1_i + s^1_i + (a^2_h + s^2_h + t^2_{hi} - a^1_i - s^1_i) z_{ih}, \ \forall \, i \in N^1, h \in N^2 : \quad a^2_h + s^2_h + t^2_{hi} > a^1_i + s^1_i \tag{4.45}$$

$$w^1_i \leq b^1_i + (b^2_h - t^2_{ih} - b^1_i) z_{ih}, \ \forall \, i \in N^1, h \in N^2 : b^2_h - t^2_{ih} < b^1_i \tag{4.46}$$

$$w^2_h \geq a^2_h + (a^1_i + t^2_{ih} - a^2_h) z_{ih}, \ \forall \, i \in N^1, h \in N^2 : a^1_i + t^2_{ih} > a^2_h \tag{4.47}$$

$$w^2_h \leq b^2_h + (T - t^1_{i(n+1)} - t^2_{hi} - s^2_h - b^2_h) z_{ih}, \ \forall \, i \in N^1, h \in N^2 : \quad T - t^1_{i(n+1)} - t^2_{hi} - s^2_h < b^2_h \tag{4.48}$$

$$w'^1_i - w^1_i \geq (t^2_{ih} + s^2_h + t^2_{hi}) z_{ih}, \ \forall \, i \in N^1, h \in N^2 \tag{4.49}$$

$$w'^1_i - w^1_i \geq s^1_i, \ \forall \, i \in N^1 \tag{4.50}$$

$$Q^2 l (w'^1_i - w^1_i) \geq \sum_{h \in N^2} \left(Q^2 s^2_h + (t^2_{ih} + t^2_{hi}) q^2_h \right) z_{ih} - M_{il} \left(1 - \sum_{j:(i,j) \in A^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x_{ij\bar{l}} \right), \ \forall \, i \in N^1, l \in L \tag{4.51}$$

$$T M_L Q^1 \sum_{j \in N^1} \sum_{l \in L} x^1_{0jl} \geq M_L \sum_{i \in N^1} \sum_{h \in N^2} (t^1_{0i} + t^1_{i0}) q^2_h z_{ih} + Q^1 \sum_{h \in N^2} \left(s^2_h + \frac{1}{Q^2} \sum_{i \in N^1} (t^2_{ih} + t^2_{hi}) q^2_h z_{ih} \right). \tag{4.52}$$

Constraints (4.30)–(4.37) are common in the literature (Dantzig; Fulkerson; Johnson, 1954; Ascheuer; Fischetti; Grötschel, 2001; Lysgaard; Letchford; Eglese, 2004; Yıldız; Karaoğlan; Altiparmak, 2023), constraints (4.38)–(4.43) are adapted for the VRPTWMDC2R from the valid inequalities proposed for the 2E-LRP by Senna et al. (2024b), and constraints (4.44)–(4.52) are novel valid inequalities proposed for this problem. Constraints (4.30) and (4.31) eliminate small subtours of two and three nodes in both vehicle and deliveryman routes. Constraints (4.32) define a lower bound on the number of vehicles used considering customer demands and vehicle capacity, and constraints (4.33) do the same for the deliverymen that leave the parking locations. Constraints (4.34)–(4.36) eliminate infeasible arcs and assignments due to time window incompatibility and deliveryman capacity. Constraint (4.37) defines that the number of parking locations visited is greater than a lower bound (P_{min}) on the number of parking locations needed to serve all customers considering their demands and the parking locations capacity. To define P_{min}, the facilities should be ordered in a decreasing lexicographic order from the one with the largest to the one with the smallest capacity. The value of P_{min} is the number of facilities obtained by following this ordered list of facilities from the one with the largest capacity

until the accumulated capacity is at least the sum of all customers demands. Constraints (4.38) state that deliverymen do not leave a parking location if it is not visited by a vehicle. Constraints (4.39) ensure that Corollary 4.1 holds. Constraints (4.40) guarantee that Proposition 4.1 holds. Constraints (4.41) state that no customer is assigned to a parking location if it is not visited by a vehicle. Constraints (4.42) define that deliverymen only leave a parking location if there are customers assigned to it. Constraints (4.43) ensure that no customer is assigned to a parking location if no deliveryman leaves it. Constraints (4.44) limit the total demand of the customers assigned to a parking location to the capacity of the deliverymen visiting this parking location. Constraints (4.45)–(4.48) define lower and upper bounds on the time variables based on the assignment of customers to parking locations. Constraints (4.49) state that the time spent in a parking location is at least the time of serving the customer that takes more time to be visited and served. Constraints (4.50) ensure that the time spent in a parking location $i \in N^1$ is greater than or equal to the lower bound s_i^1. Constraints (4.51) define the lower bound presented in Proposition 4.2 for the time spent in a parking location. In these constraints, $M_{il} = \sum_{h \in N^2}(Q^2 s_h^2 + (t_{ih}^2 + t_{hi}^2)q_h^2) - Q^2 l s_i^1$ is a sufficiently large number to ensure the validity of the constraints. Constraints (4.52) define a lower bound on the number of vehicles needed to serve the customers, based on the lower bound on the total time that the vehicles stay out of the depot from Proposition 4.4. On top of these VIs, time windows were tightened based on Ascheuer, Fischetti, and Grötschel (2001).

4.5 Benders decomposition

The VRPTWMDC2R can be decomposed in a Benders fashion (Benders, 1962; Hooker; Ottosson, 2003). Due to the high dependence of the deliveryman routes to clustering and vehicle routes, the master problem (MP) assigns customers to parking locations and defines the vehicle routes while the subproblem (SP) defines the deliveryman routes. To solve this reformulation of the VRPTWMDC2R, we design a branch-and-Benders-cut (BBC) algorithm (Moreno; Munari; Alem, 2019, 2020). Section 4.5.1 presents the MP, Section 4.5.2 introduces the SP, Section 4.5.3 discusses the BBC, Section 4.5.4 proposes some improvements to the BBC, and Section 4.5.5 introduces a mixed-integer programming (MIP) heuristic for the problem that can be used to provide a good initial solution.

4.5.1 Master Problem

Let r represent a vehicle route that starts and ends at the depot, visiting a set of parking locations to which there are customers assigned to (r represents the vehicle route and the customer assignment conjointly). Let $N_r^1 \subset N^1$ be the set of parking locations visited by this route, and $N_r^2 \subset N^2$ be the set of customers assigned to the parking locations in N_r^1. We shall represent the arcs of route r that do not connect to the depot as A_r^1. An assignment of customers to parking

locations defines sets $N_r^{[i]}, i \in N^1$, that are the sets of customers assigned to the corresponding parking location $i \in N^1$. We shall refer to a set $N_r^{[i]}$ as a cluster.

Define R as the set of all feasible pairs (r, l), in which r is a vehicle route with corresponding customer assignment, and l the number of deliverymen in this route. These pairs are all feasible with regard to constraints (4.2)–(4.29), since they represent vehicle routes and customer clustering that allow for feasible deliveryman routes. Given a pair (r, l), the cost of the deliveryman routes inside the clusters defined by $N_r^{[i]}, i \in N^1$, in the route r when traveled by a vehicle with l deliverymen is c_{rl}.

Let $\overline{R}$ be the set of infeasible pairs (r, l) when considering the deliveryman routes, i.e., the pairs that respect constraints (4.2)–(4.6), (4.14), (4.15), (4.19), (4.20), (4.22)–(4.25), and (4.29), but do not respect at least one of constraints (4.7)–(4.13), (4.16)–(4.18), (4.21), and (4.26)–(4.28). The MP is given by:

$$\text{(MP) min} \sum_{j \in N^1} \sum_{l \in L} (f^1 + lf^2) x^1_{0jl} + c^1 \sum_{(i,j) \in A^1} \sum_{l \in L} d^1_{ij} x^1_{ijl} + \sum_{i \in N^1} \eta_i \tag{4.53}$$

$$\text{s.t. (4.2)–(4.6), (4.14), (4.15), (4.19), (4.20), (4.22)–(4.25), (4.29)}$$

$$\sum_{i \in N^1_r} \eta_i \geq c_{rl} \left(\sum_{(i,j) \in A^1_r} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} + \sum_{j \in N^1_r} \sum_{h \in N^{[j]}_r} z_{jh} - |A^1_r| - |N^2_r| + 1 \right), \quad \forall\, (r, l) \in R \tag{4.54}$$

$$\sum_{(i,j) \in A^1_r} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} + \sum_{j \in N^1_r} \sum_{h \in N^{[j]}_r} z_{jh} \leq |A^1_r| + |N^2_r| - 1, \ \forall\, (r, l) \in \overline{R}. \tag{4.55}$$

The objective function (4.53) is equivalent to (4.1) with the cost of the deliveryman routes calculated based on variables η_i. Constraints (4.54) and (4.55) are optimality and feasibility cuts based on path-cuts (Parada et al., 2024; Senna et al., 2024a). We shall refer to the MP without the optimality and feasibility cuts as the relaxed MP (RMP).

To define VIs, let $R_{il} \subset R$ be the set of all assignments of customers to parking location i that are feasible considering the deliveryman routes when visited by $l \in L$ deliverymen in a back-and-forth trip from the depot (a vehicle route that only visits parking location $i \in N^1$). Accordingly, let $\overline{R}_{il} \subset \overline{R}$ be the set of all assignments of customers to parking location i that creates clusters that are infeasible when visited by $l \in L$ deliverymen in back-and-forth trips from the depot.

The MP can be strengthened by VIs (4.30), (4.32), (4.34), (4.35), (4.37), (4.40), (4.41), (4.44)–(4.46), (4.49)–(4.52). We also propose the following valid inequalities:

$$Q^2 \eta_i \geq c^2 \sum_{h \in N^2} (d^2_{ih} + d^2_{hi}) q^2_h z_{ih}, \ \forall\, i \in N^1 \tag{4.56}$$

$$\eta_i \geq c^2 (d^2_{ih} + d^2_{hi}) z_{ih}, \ \forall\, i \in N^1, h \in N^2 \tag{4.57}$$

$$\eta_i \geq c_{rM_L} \left(\sum_{h \in N_r^{[i]}} z_{ih} - |N_r^{[i]}| + 1 \right), \ \forall\, i \in N^1, (r, M_L) \in R_{iM_L} \tag{4.58}$$

$$\sum_{h \in N_r^{[i]}} z_{ih} + \sum_{j:(i,j) \in A^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} \leq |N_r^{[i]}|, \ \forall\, i \in N^1, (r, l) \in \overline{R}_{il} \tag{4.59}$$

$$w'^1_i - w^1_i \geq t_{rl} \left(\sum_{h \in N_r^{[i]}} z_{ih} + \sum_{j:(i,j) \in A^1} \sum_{\bar{l} \in L : \bar{l} \leq l} x^1_{ij\bar{l}} - |N_r^{[i]}| \right), \ \forall\, i \in N^1, (r, l) \in R_{il}. \tag{4.60}$$

Constraints (4.56) impose the lower bound presented in Proposition 4.5 for the cost of deliveryman routes inside a cluster. Constraints (4.57) state that the cost of the deliveryman routes associated to a parking location is at least the cost of visiting the farthest customer associated to it. Constraints (4.58) provide a lower bound on the cost of the deliveryman routes in a cluster. Constraints (4.59) eliminate infeasible assignments. Constraints (4.60) define a lower bound (t_{rl}) on the time spent in each parking location by the vehicle visiting it depending on the customers assigned to it and the number of deliverymen. In constraints (4.59)–(4.60), when $l = M_L$, the summation in x_{ijl} may be replaced by 1, since this is the best case scenario for costs and feasibility.

The optimality and feasibility cuts (4.54) and (4.55) and the VIs (4.58)–(4.60) are of exponential cardinality. Therefore, it is impractical to enumerate all of them a priori. Instead, one declares the RMP and starts to solve it in a branch-and-cut scheme. When a solution is found, the cuts and VIs needed for this solution are separated and included in the model. This leads to the BBC algorithm described in Section 4.5.3. The separation of cuts and VIs is made by solving the SP described next.

4.5.2 Subproblem

Given a pair $(r, l) \in R$, we define an SP that is separable by vehicle route. To simplify notation, we shall represent $N_r^{[i]}$ by $N^{[i]}$ in this context. Also, parking location i will be represented by nodes as 0_i and $n_i + 1$ for deliveryman routes source and sink, respectively, with $n_i = |N^{[i]}|$. Let $N_0^{[i]} = N^{[i]} \cup \{0_i, n_i + 1\}$. We define the complete directed graph $G^{[i]} = (N_0^{[i]}, A^{[i]})$, in which $A^{[i]} = \{(h, k) \in \widetilde{A}^2 : h, k \in N^{[i]}\} \cup (\{0_i\} : N^{[i]}) \cup (N^{[i]} : \{n_i + 1\})$.

The SP is given by

$$\text{(SP) min } c^2 \sum_{i \in N_r^1} \sum_{(h,k) \in A^{[i]}} d^2_{hk} x^2_{hk} \tag{4.61}$$

$$\text{s.t.} \sum_{h:(h,k) \in A^{[i]}} x^2_{hk} = 1, \ \forall\, k \in N^{[i]}, i \in N_r^1 \tag{4.62}$$

$$\sum_{h:(h,k) \in A^{[i]}} x^2_{hk} = \sum_{h:(k,h) \in A^{[i]}} x^2_{kh}, \ \forall\, k \in N^{[i]}, i \in N_r^1 \tag{4.63}$$

$$\sum_{h \in N^{[i]}} x^2_{0_i h} = \sum_{h \in N^{[i]}} x^2_{h(n_i+1)},\ \forall\, i \in N^1_r \tag{4.64}$$

$$\sum_{h \in N^{[i]}} x^2_{0_i h} \leq l,\ \forall\, i \in N^1_r \tag{4.65}$$

$$w^2_k \geq w^2_h + s^2_h + t^2_{hk} - M_{hk}(1 - x^2_{hk}),\ \forall\, (h,k) \in A^{[i]}, i \in N^1_r \tag{4.66}$$

$$w^2_{0_j} \geq w^2_{n_i+1} + t^1_{ij},\ \forall\, (i,j) \in A^1_r \tag{4.67}$$

$$u^2_k \geq u^2_h + q^2_k - Q^2(1 - x^2_{hk}),\ \forall\, (h,k) \in A^{[i]}, i \in N^1_r \tag{4.68}$$

$$x^2_{hk} \in \{0,1\},\ \forall\, (h,k) \in A^{[i]}, i \in N^1_r \tag{4.69}$$

$$a^2_h \leq w^2_h \leq b^2_h,\ \forall\, h \in N^{[i]}_0, i \in N^1_r \tag{4.70}$$

$$q^2_h \leq u^2_h \leq Q^2,\ \forall\, h \in N^{[i]}, i \in N^1_r. \tag{4.71}$$

The objective function (4.61) minimizes the cost of the deliveryman routes inside the clusters visited by route r. Constraints (4.62)–(4.64) are equivalent to (4.7)–(4.9) but restricted to the customers visited in the route. Constraints (4.65) limit the number of deliveryman routes in each cluster to the number of deliverymen traveling in the corresponding vehicle. Constraints (4.66) control the time flow of the deliveryman routes. Constraints (4.67) control the time flow along the vehicle route. Constraints (4.68) control the load flow inside the clusters. Constraints (4.69)–(4.71) define variable domains.

The SP can be strengthened by the following VIs:

$$\sum_{(h,k) \in A^{[i]}: h,k \in S} x^2_{hk} \leq |S| - 1,\ \forall\, i \in N^1_r, S \subset N^{[i]} : |S| \in \{2,3\} \tag{4.72}$$

$$x^2_{hk} = 0,\ \forall\, i \in N^1_r, h,k \in N^{[i]}, h \neq k : (a^2_h + s^2_h + t^2_{hk} > b^2_k) \vee (q^2_h + q^2_k > Q^2) \tag{4.73}$$

$$w^2_{n_i+1} - w^2_{0_i} \geq \max\left\{ \frac{1}{l} \sum_{h \in N^{[i]}} \left[s^2_h + (t^2_{0_i h} + t^2_{h(n_i+1)}) \frac{q^2_h}{Q^2} \right],\ \max_{h \in N^{[i]}} \left\{ t^2_{0_i h} + s^2_h + t^2_{h(n_i+1)} \right\} \right\}, \quad \forall\, i \in N^1_r. \tag{4.74}$$

Constraints (4.72) and (4.73) are equivalent to constraints (4.31) and (4.36) but restricted to the customers visited by the vehicle route. Constraints (4.74) define a lower bound on the time spent in each cluster as the maximum of the lower bound discussed in Proposition 4.2 and the time needed to serve the most time-consuming customer. Time windows are also tightened based on Ascheuer, Fischetti, and Grötschel (2001). Since in the SP the vehicle route is already defined, this tightening becomes very efficient.

The SP is used to define the optimality and feasibility cuts (4.54) and (4.55). When the SP is feasible, the value of the objective function for an optimal solution is used to define the parameter c_{rl} of constraints (4.54) for the pair $(r,l) \in R$. If the SP is not feasible, the pair $(r,l) \in \overline{R}$ and, hence, a feasibility cut (4.55) must be added to the MP.

The SP is also used to define VIs (4.58)–(4.60). To separate VIs (4.58), one defines the SP based on a vehicle route that goes from the depot to a parking location and back to the depot with M_L deliverymen. For VIs (4.59), the procedure is the same, but with the number of deliverymen that are actually traveling in the vehicle that visits the corresponding cluster (l). Finally, VIs (4.60) are separated by replacing the objective function (4.61) by $w_i'^1 - w_i^1$ and solving the SP for a route that goes back and forth from the depot to the customer $i \in N^1$ and considering the number of deliverymen $l \in L$ in the vehicle. The objective function value of an optimal solution of this problem is used to define the value of parameter t_{rl} of constraints (4.60).

4.5.3 Branch-and-Benders-cut algorithm

Due to the exponential nature of the cuts (4.54) and (4.55) and the VIs (4.58)–(4.60), it is impractical to enumerate all of them to solve the VRPTWMDC2R. Instead, we solve the problem in a branch-and-Benders-cut scheme (Moreno; Munari; Alem, 2019, 2020). To this extent, the RMP strengthened by the polynomial VIs is solved in a branch-and-cut fashion. Every time an integer solution is found, the SP is solved to separate the optimality and feasibility cuts (4.54) and (4.55) and VIs (4.58)–(4.60). The following steps summarize the BBC algorithm:

1. Declare the RMP with the polynomial VIs (4.30), (4.32), (4.34), (4.35), (4.37), (4.40), (4.41), (4.44)–(4.46), (4.49)–(4.52), (4.56), and (4.57) and start the branch-and-cut algorithm;

2. Every time a feasible integer solution is found, separate VIs (4.58)–(4.60) by solving the SP restricted to a single cluster for all clusters in the current solution. Separate also the feasibility and optimaltiy cuts (4.54) and (4.55) by solving the SP defined for all pairs (r, l) of the current solution;

3. If the current solution is feasible given the deliveryman routes, compute the overall solution cost by including the cost update given by the SP. If this cost is lower than that of the incumbent solution, update the incumbent;

4. Continue the branch-and-cut solution procedure by proceeding to the next node in the branch-and-cut tree. If a new feasible integer solution is found, return to step 2. If the time limit is reached or the optimality gap reaches the optimality tolerance, interrupt the algorithm procedure.

This algorithm can be implemented in a modern commercial solver by means of callbacks.

4.5.4 Improvements

For some instances, the separation procedures of the BBC may take a few seconds for each route, which leads to a long time spent in separation procedures, i.e., solving the MIPs that

correspond to the SP. This leads to a reduction in the rate that the branch-and-cut nodes are processed. Although essential to the BBC algorithm, it would be better to separate these cuts only when they are needed and the corresponding SP is useful, i.e., it corresponds to an important route or assignment. At the beginning of the solution procedure, the lower bound is too low and the first solutions found by the solver are usually of poor quality. Therefore, it would be interesting to separate cuts when solutions are better and the lower bound is not so low.

To overcome these issues, we propose a two-phase BBC (2P-BBC). In the first phase, only one cut is separated, enough to cut off the solution presented by the solver while reducing the computational burden of separating every possible cut. In the second phase, every VI (4.58)–(4.60) and cut (4.54)–(4.55) is separated. The second phase starts upon reaching a gap plateau, i.e., when the solution procedure remains a long time without significantly improving the optimality gap, which indicates that both the lower bound and the upper bound found by the BBC have not significantly improved. The following steps are executed in the procedure of the 2P-BBC:

1. Declare the RMP with the polynomial VIs (4.30), (4.32), (4.34), (4.35), (4.37), (4.40), (4.41), (4.44)–(4.46), (4.49)–(4.52), (4.56), and (4.57) and start the branch-and-cut algorithm;

2. Every time a feasible integer solution is found, verify whether a gap plateau has been reached. If so, go to step 5;

3. Start to separate VIs (4.58)–(4.60) by solving the SP restricted to a single cluster. Upon finding a VI that cuts off the current solution, include this VI in the model and go to step 7 without separating other VIs;

4. Start to separate cuts (4.54) and (4.55), one route at a time. Upon finding a cut that cuts off the current solution, include this cut in the model and go to step 7. If no cut has been found, go to step 6;

5. Separate all VIs (4.58)–(4.60) by solving the SP restricted to a single cluster and all feasibility and optimality cuts (4.54) and (4.55) by solving the SP defined by the vehicle routes and customer assignments of the solution;

6. If the current solution is feasible given the deliveryman routes and its cost is lower than that of the incumbent after computing the cost of the deliveryman routes, update the incumbent;

7. Continue the branch-and-cut algorithm by proceeding to the next node in the branch-and-cut tree. If a new feasible integer solution is found, return to step 2. If the time limit is reached or the optimality gap reaches the optimality tolerance, interrupt the algorithm procedure.

Like the algorithm in Section 4.5.3, this can be implemented in a commercial solver by means of callbacks. When implementing this algorithm, it is important to be careful in step 3 to ensure that, in successive callback calls, different parking locations are selected for VI separation. Otherwise, several VIs would be separated for a single node (e.g., the one with smallest index), having many cuts related to this parking location and none related to the others. This deteriorates the algorithm's performance and increases significantly the number of included constraints, most of them non-binding in an optimal solution. In our implementation, we have ordered the parking locations and defined that the first parking location to be processed in a callback call is the subsequent of the one that had a cut separated in the previous call. If no VI was separated for this parking location, the next node would be analyzed until one VI was found or it was proved that there was no VI (4.58)–(4.60) that cuts off the current solution. This way, if a VI was included for a parking location in a callback call, it would be the last one to be analyzed in the next call.

4.5.5 MIP heuristic

For some instances, the CF and the BBC showed to be slow in finding good feasible solutions for some instances. Thus, providing good initial solutions lead to better overall performance of the algorithm. This is specially important for the two-phase BBC, since the delayed separation of VIs and cuts makes it more difficult for the algorithm to update the incumbent solution at the beginning of the solution procedure.

To overcome this issue, we have developed a MIP heuristic that finds a good feasible solution in a short amount of time. The procedure is based on defining, for each customer, a list of the parking locations that have a time window opening that varies at most $0.1T$ from the moment that the customer's time window opens. The heuristic consists in solving the CF (with VIs) by limiting the parking locations to which each customer can be assigned to the α closest ones from this list. The resulting MIP is then solved by a commercial MIP solver for a few minutes or until it finds a solution with optimality gap within a tolerance. This solution is then used as a MIP start for the BBC (or the CF). If by constraining the assignment of each customer we obtain an infeasible problem, the heuristic is solved iteratively by increasing α by one until it finds a feasible solution for the problem.

4.6 Computational experiments

Computational experiments were performed to evaluate the suitability of the proposed methodology and obtain managerial insights on the problem. All algorithms were implemented in C++ and use Gurobi 11.0 solver. The optimality gap tolerance was set at 10^{-7}, the time limit at 3,600s, and the memory limit at 32GB. The experiments were performed on computers equipped with 2xAMD Rome 7532 processors running at 2.46GHz and using eight threads. For

the MIP heuristic, the optimality tolerance was set at 10%, and the time limit at 300s; the initial value of α was three.

In Section 4.6.1, the instances used in the experiments are presented. Section 4.6.2 evaluates the performance of the CF and VIs for solving the problem with the commercial solver and Section 4.6.3 does the same for the BBC. Finally, in Section 4.6.4, managerial insights are presented, shedding light onto the importance of considering both the customers clustering and the deliveryman routes in the problem.

4.6.1 Instances

Two sets of instances were used in the experiments. The first one consists of the instances proposed by Senna et al. (2024a) for the VRPTWMD with two-level routing with 50 nodes (10 parking locations and 40 customers, denoted as 10–40). These instances were generated by the authors based on the Solomon instances for the VRPTW (Solomon, 1987), having the first ten nodes of the original instance representing a parking location and randomly generating customers around them. These instances have predefined clusters, but we have ignored them for the VRPTWMDC2R.

To broaden the scope of our experiments, we have generated another set of instances with 25 nodes (five parking locations and 20 customers, referred to as 5–20). We have followed what was proposed by Senna et al. (2024a), generating customers around the parking locations represented by the original nodes of Solomon instances. The customers have time windows similar to the parking locations they are assigned to. Their coordinates follow a normal distribution with mean on the coordinate of the corresponding parking location and standard deviation $\sigma = 3$. We have generated these instances with predefined clusters as well, to follow what was proposed by Senna et al. (2024a) and because this is important for the assessment of the impact of clustering in the solution quality, as discussed in Section 4.6.4. However, we have ignored this clustering in the VRPTWMDC2R. All instances and detailed results are available at `https://www.dep.ufscar.br/munari/vrptwmd/`.

Following Senna et al. (2024a), we have considered the cost parameters to be $(f^1, c^1, f^2, c^2) = (1000, 10, 100, 1)$ and that the deliverymen travel at one third of the vehicle speed. A limit of $M_L = 3$ deliverymen in each vehicle was considered. Distances were calculated based on the euclidean distance truncated to integers. We ran the Floyd-Warshall algorithm (Cormen et al., 2009) on these distances to ensure the triangular inequality was valid. Travel times were processed accordingly. In all instances, we considered that the deliverymen capacity is 50, since it is the largest individual demand in the Solomon instances (Solomon, 1987).

4.6.2 Compact formulation

The first assessment to be made is on the performance of the MIP solver with the CF and different sets of VIs. Five different configurations were compared. The first one (CF1) corresponds to the CF (4.1)–(4.29) without VIs. The second (CF2) represents the CF with the VIs from the literature (4.30)–(4.37). The third configuration (CF3) has the CF with the VIs from the literature (4.30)–(4.37) and those that are related to the binary variables and load constraints (4.38)–(4.44). Configuration CF4 includes the CF and all VIs (4.30)–(4.52): the ones included in the other configurations and the ones related to time variables. Furthermore, the impact of using the MIP heuristic to provide a MIP start to CF4, referred to as CF4H, was evaluated.

The performance of the MIP solver with different CF configurations varies significantly depending on the instance class (C, R, or RC) to which the original Solomon instance belongs. When generating instances for the VRPTW, Solomon proposed three different classes of instances. Class “C” has its nodes separated in clusters, class “R” has the nodes uniformly randomly generated, and, for class “RC”, some of the nodes were generated as in class “C” and some as in class “R” (Solomon, 1987). Therefore, with the generation of new customers around these nodes for the VRPTWMD (Senna et al., 2024a), the geographical distribution of customers and parking locations varies significantly for different instance classes. In class “C”, there are many parking locations close to each other, having many possibly interesting parking locations to which the customers can be assigned. On the contrary, in class “R”, parking locations are usually far apart, having a little amount of candidate parking locations that are interesting for each customer. Class “RC” has an intermediate behavior.

These results are summarized in Table 4.1 for instances of size 5–20 and in Table 4.2 for those of size 10–40. The results are presented divided by instance class and aggregated by all instances as well. In these tables, for each CF configuration, “LR” represents the optimal value of the objective function of the linear programming relaxation of the VRPTWMDC2R. “LB” and “UB” stand, respectively, for the lower and upper bounds reported by the MIP solver at the ending of the solving procedure. “Gap (%)” corresponds to the optimality gap and “Time (s)” to the runtime in seconds. All these values represent the average for all instances in the corresponding classes. Moreover, “# of optimals” and “# no feasible solution” indicate, respectively, the number of instances for which the MIP solver could prove optimality for the best solution found and could not find any feasible solution. For some classes of instances, the solver could not find a feasible solution for all instances. In these cases, the corresponding values of UB and gap were reported as “N/A”. Since this only happened for instances of size 10–40, the information of “# no feasible solution” was suppressed in Table 4.1. Detailed results are available as supplementary material and also at `https://www.dep.ufscar.br/munari/vrptwmd/`.

Regarding the LR, all results indicate that the linear programming relaxation of CF1 is very weak. The inclusion of the VIs from the literature (CF2) leads to average values of LR that are over 30 times higher than those obtained for CF1 for instances of size 5–20 and over 25 times

higher for those of size 10–40. The other VIs, however, do not impact much the value of LR, representing around 1% increase in the average value from CF2 to CF4. The greater differences are observed for instances of class R, suggesting that the new VIs affect more instances with customers more spread than those with customers closer to parking locations.

Comparing CF1 and CF2 with respect to the performance of the solver while optimizing the corresponding MIP problem, it is clear that the difference in the strength of the LR directly impacts the LB and UB. For instances of size 5–20, the average LB increases 11.34% from CF1 to CF2, with the greatest difference being for class RC (13.75%). This leads to major gap improvements. In fact, the average gap is reduced by 10.72%, with the greatest improvement being for instances of class C, whose average gap goes from 12.25% to only 0.15%. The number of optimal solutions found is also increased, with 5 new ones (33.33% increase). Moreover, the average runtimes are 44.47% shorter.

Class	Metric	CF1	CF2	CF3	CF4	CF4H
C (9 instances)	LR	45	1,565	1,568	1,571	–
	LB	1,904	2,093	2,094	2,096	2,096
	UB	2,097	2,096	2,096	2,096	2,096
	Gap (%)	12.25	0.15	0.14	0.00	0.00
	Time (s)	1,926	904	562	120	69
	# of optimals	5	7	8	9	9
R (12 instances)	LR	99	2,658	2,658	2,687	–
	LB	3,718	4,122	4,121	4,220	4,220
	UB	4,220	4,220	4,220	4,220	4,220
	Gap (%)	13.30	2.60	2.62	0.00	0.00
	Time (s)	1,921	387	410	57	7
	# of optimals	8	11	11	12	12
RC (8 instances)	LR	53	1,977	1,978	1,995	–
	LB	2,443	2,779	2,784	3,906	3,906
	UB	3,941	3,911	3,911	3,911	3,911
	Gap (%)	38.81	29.62	29.48	0.14	0.13
	Time (s)	2,981	2,861	2,863	604	564
	# of optimals	2	2	2	7	7
All (29 instances)	LR	70	2,131	2,132	2,150	–
	LB	2,804	3,122	3,123	3,474	3,474
	UB	3,484	3,476	3,476	3,476	3,476
	Gap (%)	20.01	9.29	9.26	0.04	0.04
	Time (s)	2,215	1,230	1,134	228	180
	# of optimals	15	20	21	28	28

Table 4.1: Results of the CF configurations for instances of size 5–20.

For instances of size 10–40 the impact of the VIs from the literature is even greater, since the solver using CF1 cannot find feasible solutions for seven instances, while the CF2 leads to feasible solutions for all instances.

Moving on to CF3, for instances of size 5–20, there is little improvement compared to CF2. The most significant differences are that CF3 finds one extra optimal solution (for an instance of class C) and the runtimes are 37.83% shorter. For instances of size 10–40, the behavior is

similar, with the greatest improvement coming from the increase in the average LB (1.79%).

Nevertheless, CF4 presents a great improvement in the average results. For instances of size 5–20, one extra optimal solution is found for class C and one extra for class R, with all instances having an optimal solution found for these classes. For class RC, the difference from the other CFs is even greater, with 5 new optimal solutions found, having 7 out of 8 optimals found. Accordingly, the LB increases 40.30% and the UB decreases 29.34%. The runtimes are also cut by 79.89%. This indicates a great improvement in the solver performance created by the time-related VIs.

For instances of size 10–40, the behavior is similar. On average, the LB increases 42.26% compared to the CF3 value, which, combined with a 6.07% UB improvement, leads to a 28.54% gap improvement. Three new optimal solutions were also found for instances of class R.

Class	Metric	CF1	CF2	CF3	CF4	CF4H
C (9 instances)	LR	69	1,710	1,716	1,737	–
	LB	754	1,773	1,910	3,307	3,329
	UB	N/A	5,085	5,102	4,612	3,612
	Gap (%)	N/A	64.93	61.66	26.85	7.85
	Time (s)	3,600	3,600	3,600	3,600	3,601
	# of optimals	0	0	0	0	0
	# no feasible solution	2	0	0	0	0
R (12 instances)	LR	176	3,801	3,804	3,877	–
	LB	3,239	5,730	5,791	8,253	8,236
	UB	N/A	8,714	8,707	8,551	8,551
	Gap (%)	N/A	35.33	34.75	3.65	3.80
	Time (s)	3,300	3,009	3,015	2,217	1,878
	# of optimals	1	2	2	5	6
	# no feasible solution	3	0	0	0	0
RC (8 instances)	LR	101	3,955	3,962	3,964	–
	LB	1,128	4,232	4,250	5,363	5,509
	UB	N/A	9,646	9,883	8,928	8,490
	Gap (%)	N/A	55.66	56.69	39.05	34.94
	Time (s)	3,601	3,600	3,601	3,600	3,601
	# of optimals	0	0	0	0	0
	# no feasible solution	2	0	0	0	0
All (29 instances)	LR	122	3,195	3,200	3,236	–
	LB	1,885	4,089	4,162	5,921	5,961
	UB	N/A	7,845	7,913	7,433	7,001
	Gap (%)	N/A	50.13	49.16	20.62	13.65
	Time (s)	3,476	3,356	3,358	3,028	2,888
	# of optimals	1	2	2	5	6
	# no feasible solution	7	0	0	0	0

Table 4.2: Results of the CF configurations for instances of size 10–40.

The inclusion of the heuristic solution as a MIP start for the MIP solver has little effect for instances of size 5–20. For those of size 10–40, however, there is a major improvement in the solver performance, mainly for the instances of class C. In fact, for these instances, the UB is decreased by 21.68%, leading to a 19.00% gap reduction. For instances of class R, one extra

optimal solution is found, but the average LB is worse for CF4H than for CF4. For instances of class RC, there are improvements in the LB and UB that lead to a 4.11% gap improvement. On the overall average for instances of size 10–40, there is a 6.97% gap reduction.

Comparing the solver performance for different instance classes, it is clear that RC instances are the hardest ones to solve, being the only class that does not have an optimal solution found for all instances of size 5–20 and the class with largest average gap for size 10–40. Moreover, class R has the instances that presented the best results for the solver with the proposed formulations. In fact, for the ones with size 5–20, there is no significant difference between class C and class R since optimal solutions were found for all instances of both of these instance classes. However, for those with size 10–40, the solver has a much better performance for class R than for class C, since it finds six optimal solutions for class R and none for class C. The average gap is also better for R than for C.

These results show the positive impact encompassed by the proposed valid inequalities and the use of the MIP heuristic to provide a MIP start. The solver under configuration CF1 (without VIs) has a poor performance due to its very weak LR. In fact, the average gap for instances of size 5–20 is over 20% and, for those of size 10–40, it cannot even find a feasible solution for 24.14% of the instances. When including all VIs and the heuristic (CF4H), for instances of size 5–20, the gap drops to only 0.04% with only one instance not having an optimal solution found. The runtimes are also greatly reduced (91.87% reduction in the average value). For those of size 10–40, on top of all instances having a feasible solution found by the solver, six of them also have an optimal solution found and the LB increases by 216.23%.

4.6.3 Branch-and-Benders-cut

In this section, we compare the performance of different configurations of the BBC algorithm. Since our experiments with VIs show that all of them are beneficial for the solver performance, we have included all presented VIs in the BBC. More specifically, VIs (4.30), (4.32), (4.34), (4.35), (4.37), (4.40), (4.41), (4.44)–(4.46), (4.49)–(4.52), (4.56), and (4.57) were included in the MP, and VIs (4.72)–(4.74) in the SP. The first configuration is BBC1, which represents the BBC with all polynomial VIs but without the exponential VIs (4.58)–(4.60). BBC2 corresponds to the BBC1 with the inclusion of VIs (4.58)–(4.60). BBC2H includes in the BBC2 the MIP heuristic solution as a MIP start. Finally, 2P-BBC2H has also the two-phase scheme discussed in Section 4.5.4. The results are presented in Table 4.3 for instances of size 5–20 and Table 4.4 for those of size 10–40. These tables also include the results for CF4 and CF4H for comparison, since they were the ones with the best performance among the CF configurations.

Comparing the BBC1 with the BBC2, their results vary depending on the instance class and size. For those of class R, they have equivalent behaviors for both size 5–20 and 10–40. For instances of class C, the BBC2 outperforms the BBC1 for both sizes. However, for class RC, the BBC1 presents the best results for size 5–20, having a gap that is 5.48% smaller, while, for

Class	Metric	CF4	CF4H	BBC1	BBC2	BBC2H	2P-BBC2H
C (9 instances)	LB	2,096	2,096	2,084	2,089	2,088	2,089
	UB	2,096	2,096	2,097	2,097	2,098	2,097
	Gap (%)	0.00	0.00	0.83	0.52	0.61	0.54
	Time (s)	120	69	1,627	1,528	1,578	1,566
	# of optimals	9	9	5	6	6	6
R (12 instances)	LB	4,220	4,220	4,220	4,220	4,220	4,220
	UB	4,220	4,220	4,220	4,220	4,220	4,220
	Gap (%)	0.00	0.00	0.00	0.00	0.00	0.00
	Time (s)	57	7	7	26	16	16
	# of optimals	12	12	12	12	12	12
RC (8 instances)	LB	3,906	3,906	3,897	3,694	3,911	3,911
	UB	3,911	3,911	3,912	3,911	3,911	3,911
	Gap (%)	0.14	0.13	0.38	5.86	0.00	0.00
	Time (s)	604	564	739	642	346	362
	# of optimals	7	7	7	7	8	8
All (29 instances)	LB	3,474	3,474	3,468	3,413	3,473	3,473
	UB	3,476	3,476	3,476	3,476	3,476	3,476
	Gap (%)	0.04	0.04	0.36	1.78	0.19	0.17
	Time (s)	228	180	711	662	592	592
	# of optimals	28	28	24	25	26	26

Table 4.3: Results of the BBCs for instances of size 5–20.

size 10–40, the BBC2 outperforms BBC1 since BBC1 is unable to find a feasible solution for one instance. One interesting result is that, for class R, both of them find an optimal solution for all instances, greatly outperforming the CFs, that only find optimal solutions for half of the instances of class R and size 10–40.

Upon the inclusion of the MIP heuristic (BBC2H), for instances of class RC and size 5–20, there is a great improvement, since all instances have an optimal solution found, with the average runtime being reduced by 46.11% compared to BBC2 and by 53.18% compared to BBC1. On average, for size 5–20, BBC2H has an LB that is 1.76% greater than that of BBC2 and a gap that is 0.17% smaller than that of BBC1 and 1.59 smaller that the one of BBC2.

The greatest improvements created by the heuristic, however, are for instances of size 10–40. In fact, for those of class C, the UB is reduced by 25.40% compared to the BBC2, leading to a 19.26% improvement in the average gap. For instances of class RC, the behavior is similar, with a 15.68% gap reduction. On average, the UB of BBC2H is 10.89% lower than that of BBC2, and the gap is improved by 10.31%. There is also one extra optimal solution found.

The 2P-BBC2H leads to an additional 0.02% gap improvement for instances of class 5–20. The greater advantage of the two-phase procedure, however, is for instances of size 10–40. For those of class C, there is an LB improvement that leads to a 0.11% gap reduction compared to BBC2H. For class RC, the LB is improved by 0.64%, the UB by 1.85%, and the gap by 1.10%. On average, for instances of size 10–40, the 2P-BBC2H provides a 0.33% gap improvement compared to BBC2H.

Comparing the 2P-BBC2H with the CF4H, for instances of size 5–20, the average LB of

the 2P-BBC2H is 0.03% lower than that of the CF4H, the average UBs are the same, and the gap is 0.13% lower for the CF4H. This suggests that the quality of the solutions found by these configurations is practically the same, although the CF4H is more efficient in proving optimality (there are two extra optimal solutions found by the CF4H). For instances of size 10–40, however, the 2P-BBC2H outperforms the solver with the CFs. In fact, the average LB of the 2P-BBC2H is 2.37% higher than that of the CF4H and the UB is 1.14% lower, leading to a 1.89% gap reduction. Moreover, the average runtime is 23.20% shorter in the 2P-BBC2H and the 2P-BBC2H finds an optimal solution for 7 extra instances (a 116.67% increase compared to the CF4H).

Class	Metric	CF4	CF4H	BBC1	BBC2	BBC2H	2P-BBC2H
C (9 instances)	LB	3,307	3,329	3,240	3,260	3,269	3,275
	UB	4,612	3,612	4,902	4,854	3,621	3,624
	Gap (%)	26.85	7.85	32.23	29.03	9.77	9.66
	Time (s)	3,600	3,601	3,605	3,604	3,605	3,604
	# of optimals	0	0	0	0	0	0
	# no feasible solution	0	0	0	0	0	0
R (12 instances)	LB	8,253	8,236	8,539	8,539	8,539	8,539
	UB	8,551	8,551	8,539	8,539	8,539	8,539
	Gap (%)	3.65	3.80	0.00	0.00	0.00	0.00
	Time (s)	2,217	1,878	68	204	251	276
	# of optimals	5	6	12	12	12	12
	# no feasible solution	0	0	0	0	0	0
RC (8 instances)	LB	5,363	5,509	5,280	5,060	5,592	5,628
	UB	8,928	8,490	N/A	10,060	8,360	8,205
	Gap (%)	39.05	35.00	N/A	48.53	32.85	31.75
	Time (s)	3,600	3,601	3,604	3,605	3,597	3,573
	# of optimals	0	0	0	0	1	1
	# no feasible solution	0	0	1	0	0	0
All (29 instances)	LB	5,921	5,961	5,995	5,941	6,091	6,102
	UB	7,433	7,001	N/A	7,815	6,964	6,921
	Gap (%)	20.62	13.65	N/A	22.40	12.09	11.76
	Time (s)	3,028	2,888	2,141	2,198	2,215	2,218
	# of optimals	5	6	12	12	13	13
	# no feasible solution	0	0	1	0	0	0

Table 4.4: Results of the BBCs for instances of size 10–40.

All these results show that both the proposed VIs and the BBCs significantly improve the performance of the MIP solver, since the CF1 (without VIs) presents a very poor performance when solved by the MIP solver. The proposed VIs, lower bounds, and heuristic lead to a much better performance while maintaining the formulation compact (i.e., with a polynomial number of variables and constraints). The Benders decomposition also leads to a very good performance when applied in one of the proposed BBCs, although generating a formulation with an exponential number of constraints. For the smaller instances, the solver under configuration CF4H slightly outperforms the BBCs, but, for the larger ones, the 2P-BBC2H clearly has the best performance among all developed methods.

4.6.4 Managerial insights

As discussed in Section 4.2, the VRPTWMD was proposed by Pureza, Morabito, and Reimann (2012) with two simplifying hypotheses: (*i*) the customer clusters can be predefined, and (*ii*) the deliveryman routes can be preprocessed. Senarclens de Grancy and Reimann (2015) and Senarclens de Grancy (2015) extended the problem by relaxing hypothesis (*i*), while keeping hypothesis (*ii*), resulting in the VRPTWMD with customer clustering (VRPTWMDC). Senna et al. (2024a) also extended the VRPTWMD by relaxing hypothesis (*ii*), while keeping hypothesis (*i*), creating the VRPTWMD with two-level routing (VRPTWMD2R). In this paper, we have proposed the VRPTWMDC2R, which relaxes both of these hypotheses.

In the previous sections, we have discussed the performance of the proposed methodologies to solve the VRPTWMDC2R. It is important, however, to assess the relevance of including the decisions of clustering and deliveryman routes in the problem, i.e., the impact on the solution quality of the VRPTWMDC2R compared to the other VRPTWMD variants. To this extent, we have performed some experiments simulating the different variants. As discussed in Section 4.6.1, the instances used in the experiments have predefined clusters that were ignored in the VRPTWMDC2R. Nevertheless, these clusters were used to simulate the VRPTWMD and the VRPTWMD2R, by setting the variables z_{ih} to match the assignment provided by the instance. Furthermore, to simulate the preprocessing of deliveryman routes, we have assumed that, instead of making direct trips between customers, in both the VRPTWMD and the VRPTWMDC, the deliveryman must always come back to the vehicle to take more goods as discussed in Section 4.2. This can be done by redefining the distances (travel times) between customers as the distances (travel times) passing through the parking location instead of the euclidean distances. Finally, to further evaluate the impact of clustering, we have extended our analysis by generating instances with varying customer dispersion. To this extent, we have followed the procedure proposed by Senna et al. (2024a) and discussed in Section 4.6.1, but generated instances with different values of standard deviation for the customers coordinates ($\sigma \in \{1, 3, 5\}$). This way, there are instances with the customers closer to ($\sigma = 1$) and farther from ($\sigma = 5$) the parking locations.

The results of these experiments are presented in Table 4.5 for a subset of instances to which it was possible to prove optimality for all configurations. In this table, on top of presenting the results divided by instance class and standard deviation σ of customer's coordinates, the aggregated total is also provided. "Cost" represents the overall solution cost, "# of veh." corresponds to the number of vehicles used in the solution, "# of del." indicates the size of the deliveryman crew used, "veh. dist." stands for the distance traveled by the vehicles, and "del. dist." shows the distance traveled by the deliverymen. All presented values are computed as averages.

The results clearly indicate that, the higher the σ, the higher the difference between the solutions of the variants and, hence, the greater the importance including the clustering and the deliveryman routes in the optimization. As expected, the costs of the VRPTWMD are the

Class	σ	Metric	VRPTWMD	VRPTWMDC	VRPTWMD2R	VRPTWMDC2R
C	1	Cost	2,057	2,049	2,039	2,027
		# of veh.	1.33	1.33	1.33	1.33
		# of del.	1.33	1.33	1.33	1.33
		Veh. dist.	54.00	52.67	54.00	52.67
		Del. dist.	50.67	55.33	32.33	34.00
	3	Cost	2,225	2,168	2,171	2,089
		# of veh.	1.33	1.33	1.33	1.33
		# of del.	2.00	1.67	2.00	1.33
		Veh. dist.	53.67	53.00	53.67	53.67
		Del. dist.	154.67	138.00	100.67	86.00
	5	Cost	2,351	2,297	2,209	2,173
		# of veh.	1.33	1.33	1.33	1.33
		# of del.	2.33	2.00	1.67	1.67
		Veh. dist.	54.00	54.00	54.00	53.67
		Del. dist.	244.00	224.00	169.00	136.67
R	1	Cost	3,732	3,732	3,672	3,672
		# of veh.	2.00	2.00	2.00	2.00
		# of del.	2.33	2.33	2.00	2.00
		Veh. dist.	144.33	144.33	144.00	144.00
		Del. dist.	55.33	55.33	32.33	32.33
	3	Cost	3,952	3,952	3,803	3,803
		# of veh.	2.00	2.00	2.00	2.00
		# of del.	4.00	4.00	3.00	3.00
		Veh. dist.	141.67	141.67	141.67	141.67
		Del. dist.	135.33	135.33	86.33	86.33
	5	Cost	5,569	5,569	4,891	4,891
		# of veh.	3.00	3.00	2.67	2.67
		# of del.	7.67	7.67	5.00	5.00
		Veh. dist.	156.67	156.67	155.67	155.67
		Del. dist.	236.00	236.00	168.00	168.00
RC	1	Cost	2,114	2,112	2,028	2,027
		# of veh.	1.00	1.00	1.00	1.00
		# of del.	2.00	2.00	1.33	1.33
		Veh. dist.	86.33	86.33	86.33	86.33
		Del. dist.	50.67	48.67	31.67	30.67
	3	Cost	4,109	4,102	3,972	3,964
		# of veh.	2.00	2.00	2.00	2.00
		# of del.	4.33	4.00	3.67	3.33
		Veh. dist.	153.67	157.00	151.00	154.33
		Del. dist.	138.67	132.00	95.00	87.67
	5	Cost	6,151	4,934	5,312	4,176
		# of veh.	3.00	2.33	2.67	2.00
		# of del.	6.67	6.00	4.67	4.33
		Veh. dist.	225.33	178.00	201.33	159.00
		Del. dist.	231.33	220.67	165.67	153.00
Total		Cost	3,584	3,435	3,344	3,203
		# of veh.	1.89	1.81	1.81	1.74
		# of del.	3.63	3.44	2.74	2.59
		Veh. dist.	118.85	113.74	115.74	111.22
		Del. dist.	144.07	138.37	97.89	90.52

Table 4.5: The impact of different VRPTWMD variants on the solution quality.

highest, since it is the most constrained variant, and those of the VRPTWMDC2R are the lowest, since it is the least constrained variant. In general, the costs of VRPTWMD2R are lower than those of the VRPTWMDC, but not always, and, in the total average, the solution cost of the VRPTWMD2R is higher than that of the VRPTWMDC. Furthermore, the distance traveled by the deliverymen is always smaller for the variants that optimize the deliveryman routes than for those that consider them to be approximated a priori.

For instances of class C, an interesting result is that the vehicle distances are the same for the VRPTWMD and the VRPTWMD2R, suggesting that the vehicle routes do not change. Looking in more detail, for instances with $\sigma = 1$, the difference in the solutions is concentrated in the distance traveled by vehicles and deliverymen, with little impact in the overall cost. For those with $\sigma = 3$ and $\sigma = 5$, however, differences in the number of deliverymen also appear, although the size of the vehicle fleet is always the same. As a consequence, the deliveryman routes distance is significantly impacted, having a 43.99% reduction from the VRPTWMD to the VRPTWMDC2R in the instances with $\sigma = 5$, leading to a 7.57% cost reduction. The cost improvement of the VRPTWMDC2R compared to the VRPTWMDC is 5.40% and to the VRPTWMD2R is 1.63% for those instances.

For instances of class R, the solutions of the VRPTWMD and the VRPTWMDC are always the same. Accordingly, the solutions of the VRPTWMD2R and the VRPTWMDC2R are also the same. This suggests that, for these instances, the clustering can be easily preprocessed with little impact on the solution. This happens because these instances have their parking locations very far apart and, hence, clustering becomes trivial. The impact of considering the deliveryman routes, however, is not negligible. In fact, this leads to a reduction on the size of the deliveryman crew that ranges from 14.16% for instances with $\sigma = 1$ to 34.81% for those with $\sigma = 5$. Accordingly, the reduction on the distance traveled by the deliverymen ranges from 28.81% for instances with $\sigma = 5$ to 41.57% for those with $\sigma = 1$. Combined with some reductions in the vehicle fleet size and the distance traveled by the vehicles, this leads to a cost reduction that goes from 1.61% for instances with $\sigma = 1$ to 12.17% for those with $\sigma = 5$.

For instances of class RC, the number of vehicles used is the same for all variants and instances with $\sigma = 1$ and $\sigma = 3$. For instances with $\sigma = 5$, however, the reduction from the VRPTWMD to the VRPTWMDC2R is of 33.33%. For instances with $\sigma = 1$, the main difference among the solutions of the different variants is on the distance traveled by the deliverymen, which is significantly reduced by the variants that optimize these routes. This leads to the reduction on the average solution cost of 4.12% from the VRPTWMD to the VRPTWMDC2R. The clustering, however, has little impact. The behavior for $\sigma = 3$ is similar. Nonetheless, for instances with $\sigma = 5$, the clustering has a huge impact. In fact, the cost reduction from the VRPTWMD to the VRPTWMDC is of 19.79% while from the VRPTWMD2R to the VRPTWMDC2R it is of 21.39%. Combined with the cost reduction obtained by the inclusion of the deliveryman routes in the optimization, this leads to an average solution cost of the VRPTWMDC2R that is 32.11% smaller than that of the VRPTWMD.

Considering the overall average of instances, it is clear that the optimization of the deliveryman routes is of major relevance in the optimal size of the deliveryman crew and the distance traveled by them. Moreover, the clustering is important depending on the instances characteristics, as expected. On the one hand, for instances that have parking locations far apart from each other and customers close to them, the clustering becomes trivial and, hence, its optimization is not impactful. On the other hand, for instances that have many interesting candidate parking locations for each customer, the clustering becomes relevant and its optimization may lead to major cost reductions. On the overall average, the VRPTWMDC2R has a solution cost that is 10.63% smaller than that of the VRPTWMD.

4.7 Conclusion

In this paper, we have introduced a variant of the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD). This problem emulates a last-mile delivery scheme that has vehicles traveling with more than one deliveryman to increase the number of customers that can be served with a single stop of the vehicle while reducing the overall time that the vehicles stay parked throughout the route. Since deliveryman costs and GHGs emissions are usually smaller than those of the vehicles, this allows for a cheaper and greener delivery system.

As originally proposed, the VRPTWMD considers that the decision on which customers are to be served by each stop of the vehicles (clustering) and the deliveryman routes inside the clusters can be defined in a preprocessing phase. In previous studies, the problem had been extended to encompass either the clustering or the deliveryman routes in the optimization, but never both. With this paper, we have bridged this gap by introducing the vehicle routing problem with time windows, multiple deliverymen, customer clustering, and two-level routing (VRPTWMDC2R), which is a rich vehicle routing problem with applications in last-mile delivery.

We have formally defined the VRPTWMDC2R by means of a mathematical formulation. Theoretical properties and lower bounds have been discussed and used to propose valid inequalities. The problem has also been decomposed in a Benders fashion to develop a branch-and-Benders-cut algorithm to solve it. Computational experiments show the suitability of the proposed methodology to solve the problem.

Furthermore, managerial insights were provided to shed light onto the importance of optimizing the customer clustering and the deliveryman routes in the VRPTWMD. Our results show that the optimization of deliveryman routes is always beneficial. The clustering, however, depends on the instance characteristics. In fact, if customers are closely distributed around parking locations that are far apart from each other, clustering becomes trivial and its optimization is not relevant. Nevertheless, in situations that have many parking locations close to each other with customers distributed around them without an obvious clustering pattern, optimizing the customer clusters is of major relevance.

Finally, some possibilities of future work are the proposition of variants that include the pos-

sibility of having the deliverymen coming back to the vehicles at different parking locations or even changing vehicles throughout the route. Other interesting extensions would be the study of the problem under uncertainties (e.g., in the demand or travel time) by means of robust or stochastic optimization. New methods based on metaheuristics could also provide better solutions for large scale instances.

5 Conclusion

In this dissertation, new developments regarding two-echelon routing problems for last-mile delivery were presented. In particular, the two-echelon location-routing problem (2E-LRP) and the vehicle routing problem with time windows and multiple deliverymen (VRPTWMD) were addressed. As were the objectives of this dissertation, the literature on the 2E-LRP and the VRPTWMD was reviewed, novel formulations for the 2E-LRP were introduced, two realistic extensions for the VRPTWMD were proposed and formulated, valid inequalities and exact solution methods for these variants were presented, and the benefits of these new variants on the solution quality of the VRPTWMD were assessed. Hence, all the objectives were attained.

In particular, regarding the 2E-LRP, we have proposed novel formulations based on two-index arc variables, contrasting with the benchmark formulation from the literature, which is based on variables with a vehicle index. Novel and literature-based formulations were also evaluated, and their linear programming relaxations were compared. Extensive computational experiments have shown that the proposed formulations greatly outperform the benchmark one. Moreover, 125 new best known lower bounds were discovered, as well as 55 new optimal solutions. These developments have been discussed in Chapter 2, which is a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos). This paper is publicly available at the CIRRELT repository and is under revision for publication at a renowned Operations Research journal (Senna et al., 2024b).

Furthermore, two realistic extensions of the VRPTWMD were introduced. The first one is the VRPTWMD with two-level routing (VRPTWMD2R). It was formally defined and formulated. Several valid inequalities were proposed, as well as a Benders decomposition to craft a branch-and-Benders-cut algorithm. The proposed algorithm has shown to be very efficient to solve the problem, since it has proved optimality to 88.97% of the instances, many of them of realistic sizes. Moreover, managerial insights were provided. First, we showed the importance of properly evaluating the deliveryman routes, which in our experiments showed to give solutions that were around 10% better than the ones that ignore these routes (following the literature). Sensitivity analyses on costs were also performed and possibilities of further improvement in the use of deliverymen were discussed. We have highlighted the opportunities for cost reduction enabled by the business model encompassed by the problem, while having the beneficial side effect of reducing the emission of greenhouse gases and other pollutants due to the reduction in

the use of vehicles and the distance traveled by them. These results were presented in Chapter 3: a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos). This paper has been published at the European Journal of Operational Research (Senna et al., 2024a).

The second extension is the VRPTWMD with customer clustering and two-level routing (VRPTWMDC2R), which encompasses both the deliveryman routes and the definition of clusters of customers that are served with a single stop of the vehicle. The problem was formally defined and formulated. Theoretical properties were discussed and useful lower bounds were proposed. Several valid inequalities were introduced. A Benders decomposition-based exact algorithm was proposed. The computational experiments show that the proposed developments greatly improve the solver performance. Moreover, the relevance of including the deliveryman routes and the customer clustering in the optimization problem was thoroughly discussed. These outcomes have been introduced in Chapter 4, which is a paper coauthored with Prof. Leandro C. Coelho (Université Laval), Prof. Reinaldo Morabito (Federal University of São Carlos), and Prof. Pedro Munari (Federal University of São Carlos). This paper is on its final stages of preparation and should be submitted for publication soon.

In conclusion, the main contributions of this dissertation were novel developments considering two-echelon routing problems in last-mile delivery. In particular, for the 2E-LRP, two formulations based on two-index arc variables and valid inequalities were proposed, and 125 new best known lower bounds and 55 new optimal solutions were found for benchmark instances. For the VRPTWMD, two realistic variants were introduced and formulated, with the proposition of valid inequalities and branch-and-Benders-cut algorithms to solve these problems. The computational experiments show the suitability of the proposed formulation to solve the problem and the impact on the solution cost upon considering these variants compared to the literature-based VRPTWMD (10% cost reduction on average).

Finally, there are great possibilities for future work based on the research documented in this dissertation. Considering the 2E-LRP, for example, the proposed formulations could be extended to reflect the many variants available in the literature, or exact methods based on branch-and-cut schemes and other ad hoc solution methods could further improve the best known lower and upper bounds for the benchmark instances from the literature. Regarding the VRPTWMD, new extensions considering that the deliveryman routes could end in parking locations that are different from the ones they departed from or the deliveryman could change vehicles throughout the route are interesting extensions for the problem that reflect realistic applications. The study of the problem considering uncertainties in some of the parameters or the development of metaheuristics could also lead to interesting results.

6 References

Agnimo, V.; Ouhimmou, M.; Paquet, M.; Montecinos, J. Integrated strategic and tactical design of multi-echelon city distribution systems with vehicles synchronization: A case of the Greater Montréal area. **Computers & Industrial Engineering**, v. 183, p. 109458, 2023.

Alfandari, L.; Ljubić, I.; De Melo da Silva, M. A tailored Benders decomposition approach for last-mile delivery with autonomous robots. **European Journal of Operational Research**, v. 299, n. 2, p. 510–525, 2022.

Álvarez, A.; Munari, P. An exact hybrid method for the vehicle routing problem with time windows and multiple deliverymen. **Computers & Operations Research**, v. 83, p. 1–12, 2017.

_____. Metaheuristic approaches for the vehicle routing problem with time windows and multiple deliverymen. **Gestão e Produção**, v. 23, n. 2, p. 279–293, 2016.

Amiri, M.; Amin, S. H.; Tavakkoli-Moghaddam, R. A Lagrangean decomposition approach for a novel two-echelon node-based location-routing problem in an offshore oil and gas supply chain. **Transportation Research Part E: Logistics and Transportation Review**, v. 128, p. 96–114, 2019.

Ascheuer, N.; Fischetti, M.; Grötschel, M. Solving the asymmetric travelling salesman problem with time windows by branch-and-cut. **Mathematical Programming**, v. 90, p. 475–506, 2001.

Bayliss, C.; Bektaş, T.; Tjon-Soei-Len, V.; Rohner, R. Designing a multi-modal and variable-echelon delivery system for last-mile logistics. **European Journal of Operational Research**, v. 307, n. 2, p. 645–662, 2023.

Bektaş, T.; Laporte, G. The pollution-routing problem. **Transportation Research Part B: Methodological**, v. 45, n. 8, p. 1232–1250, 2011.

Belgin, O.; Karaoğlan, I.; Altiparmak, F. Two-echelon vehicle routing problem with simultaneous pickup and delivery: Mathematical model and heuristic approach. **Computers & Industrial Engineering**, v. 115, p. 1–16, 2018.

Ben Mohamed, I.; Klibi, W.; Sadykov, R.; Şen, H.; Vanderbeck, F. The two-echelon stochastic multi-period capacitated location-routing problem. **European Journal of Operational Research**, v. 306, n. 2, p. 645–667, 2023.

Benders, J. F. Partitioning procedures for solving mixed-variables programming problems. **Numerische Mathematik**, INFORMS, v. 4, n. 1, p. 238–252, 1962.

Boccia, M.; Crainic, T. G.; Sforza, A.; Sterle, C. **Location-routing models for designing a two-echelon freight distribution system**. [S.l.: s.n.], 2011. Technical report CIRRELT-2011-06, Université de Montréal.

Boysen, N.; Fedtke, S.; Schwerdfeger, S. Last-mile delivery concepts: A survey from an operational research perspective. **OR Spectrum**, Springer, v. 43, p. 1–58, 2021.

Breunig, U.; Schmid, V.; Hartl, R. F.; Vidal, T. A large neighbourhood based heuristic for two-echelon routing problems. **Computers & Operations Research**, v. 76, p. 208–225, 2016.

Cabrera, N.; Cordeau, J.-F.; Mendoza, J. E. The doubly open park-and-loop routing problem. **Computers & Operations Research**, v. 143, p. 105761, 2022.

Cheng, C.; Zhu, R.; Costa, A. M.; Thompson, R. G.; Huang, X. Multi-period two-echelon location routing problem for disaster waste clean-up. **Transportmetrica A: Transport Science**, v. 18, n. 3, p. 1053–1083, 2022.

Codato, G.; Fischetti, M. Combinatorial Benders' cuts for mixed-integer linear programming. **Operations Research**, v. 54, n. 4, p. 756–766, 2006.

Contardo, C.; Hemmelmayr, V.; Crainic, T. G. Lower and upper bounds for the two-echelon capacitated location-routing problem. **Computers & Operations Research**, v. 39, n. 12, p. 3185–3199, 2012.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; Stein, C. **Introduction to Algorithms**. 4. ed. Boston, USA: The MIT Press, 2009.

Cuda, R.; Guastaroba, G.; Speranza, M. A survey on two-echelon routing problems. **Computers & Operations Research**, v. 55, p. 185–199, 2015.

Dai, Z.; Aqlan, F.; Gao, K.; Zhou, Y. A two-phase method for multi-echelon location-routing problems in supply chains. **Expert Systems with Applications**, v. 115, p. 618–634, 2019.

Dantzig, G.; Fulkerson, R.; Johnson, S. Solution of a large-scale traveling-salesman problem. **Journal of the Operations Research Society of America**, INFORMS, v. 2, n. 4, p. 393–410, 1954.

Dantzig, G. B.; Ramser, J. H. The truck dispatching problem. **Management Science**, INFORMS, v. 6, n. 1, p. 80–91, 1959.

Darvish, M.; Archetti, C.; Coelho, L. C.; Speranza, M. G. Flexible two-echelon location routing problem. **European Journal of Operational Research**, v. 277, n. 3, p. 1124–1136, 2019.

De La Vega, J.; Munari, P.; Morabito, R. Exact approaches to the robust vehicle routing problem with time windows and multiple deliverymen. **Computers & Operations Research**, v. 124, p. 105062, 2020.

De La Vega, J.; Munari, P.; Morabito, R. Robust optimization for the vehicle routing problem with multiple deliverymen. **Central European Journal of Operations Research**, v. 27, n. 4, p. 905–936, 2019.

Dolan, E. D.; Moré, J. J. Benchmarking optimization software with performance profiles. **Mathematical Programming**, v. 91, p. 201–213, 2002.

Drexl, M.; Schneider, M. A survey of variants and extensions of the location-routing problem. **European Journal of Operational Research**, v. 241, n. 2, p. 283–308, 2015.

Enthoven, D. L. J. U.; Jargalsaikhan, B.; Roodbergen, K. J.; uit het Broek, M. A. J.; Schrotenboer, A. H. The two-echelon vehicle routing problem with covering options: City logistics with cargo bikes and parcel lockers. **Computers & Operations Research**, v. 118, p. 104919, 2020.

Escobar-Vargas, D.; Crainic, T. G. Multi-attribute two-echelon location routing: Formulation and dynamic discretization discovery approach. **European Journal of Operational Research**, v. 314, n. 1, p. 66–78, 2024.

Friedrich, C.; Elbert, R. Adaptive large neighborhood search for vehicle routing problems with transshipment facilities arising in city logistics. **Computers & Operations Research**, v. 137, p. 105491, 2022.

Furtado, M. G. S.; Munari, P.; Morabito, R. Pickup and delivery problem with time windows: A new compact two-index formulation. **Operations Research Letters**, v. 45, n. 4, p. 334–341, 2017.

Gavish, B.; Graves, S. C. **The traveling salesman problem and related problems**. [S.l.: s.n.], 1978. Technical report, Operations Research Center, Massachusetts Institute of Technology. OR 078-78.

Govindan, K.; Jafarian, A.; Khodaverdi, R.; Devika, K. Two-echelon multiple-vehicle location-routing problem with time windows for optimization of sustainable supply chain network of perishable food. **International Journal of Production Economics**, v. 152, p. 9–28, 2014.

Haimovich, M.; Kan, A. H. G. R. Bounds and heuristics for capacitated routing problems. **Mathematics of Operations Research**, INFORMS, v. 10, n. 4, p. 527–542, 1985.

Hooker, J. N.; Ottosson, G. Logic-based Benders decomposition. **Mathematical Programming**, Springer, v. 96, n. 1, p. 33–60, 2003.

Jacobsen, S. K.; Madsen, O. B. G. A comparative study of heuristics for a two-level routing-location problem. **European Journal of Operational Research**, v. 5, n. 6, p. 378–387, 1980.

Le Colleter, T.; Dumez, D.; Lehuédé, F.; Péton, O. Small and large neighborhood search for the park-and-loop routing problem with parking selection. **European Journal of Operational Research**, v. 308, n. 3, p. 1233–1248, 2023.

Li, H.; Chen, J.; Wang, F.; Bai, M. Ground-vehicle and unmanned-aerial-vehicle routing problems from two-echelon scheme perspective: A review. **European Journal of Operational Research**, v. 294, n. 3, p. 1078–1095, 2021.

Li, H.; Wang, H.; Chen, J.; Bai, M. Two-echelon vehicle routing problem with satellite bi-synchronization. **European Journal of Operational Research**, v. 288, n. 3, p. 775–793, 2021.

Lysgaard, J.; Letchford, A. N.; Eglese, R. W. A new branch-and-cut algorithm for the capacitated vehicle routing problem. **Mathematical Programming**, v. 100, p. 423–445, 2004.

Madsen, O. B. G. Methods for solving combined two level location-routing problems of realistic dimensions. **European Journal of Operational Research**, v. 12, n. 3, p. 295–301, 1983.

Martinez-Sykora, A.; Lamas-Fernandez, C.; Bektaş, T.; Cherrett, T.; Allen, J. Optimised solutions to the last-mile delivery problem in London using a combination of walking and driving. **Annals of Operations Research**, v. 295, p. 645–693, 2020.

Miller, C. E.; Tucker, A. W.; Zemlin, R. A. Integer programming formulation of traveling salesman problems. **Journal of the Association for Computing Machinery**, v. 7, n. 4, p. 326–329, 1960.

Mirhedayatian, S. M.; Crainic, T. G.; Guajardo, M.; Wallace, S. W. A two-echelon location-routing problem with synchronisation. **Journal of the Operational Research Society**, v. 72, n. 1, p. 145–160, 2021.

Moreno, A.; Munari, P.; Alem, D. A branch-and-Benders-cut algorithm for the crew scheduling and routing problem in road restoration. **European Journal of Operational Research**, v. 275, n. 1, p. 16–34, 2019.

_____. Decomposition-based algorithms for the crew scheduling and routing problem in road restoration. **Computers & Operations Research**, Elsevier, v. 119, p. 104935, 2020.

Moshref-Javadi, M.; Winkenbach, M. Applications and research avenues for drone-based models in logistics: A classification and review. **Expert Systems with Applications**, v. 177, p. 114854, 2021.

Munari, P.; Morabito, R. A branch-price-and-cut algorithm for the vehicle routing problem with time windows and multiple deliverymen. **TOP**, v. 26, n. 3, p. 437–464, 2018.

Munari, P.; Savelsbergh, M. Compact formulations for split delivery routing problems. **Transportation Science**, v. 56, n. 4, p. 1022–1043, 2022.

Murray, C. C.; Chu, A. G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. **Transportation Research Part C: Emerging Technologies**, v. 54, p. 86–109, 2015.

Nguyen, V.-P.; Prins, C.; Prodhon, C. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. **Engineering Applications of Artificial Intelligence**, v. 25, n. 1, p. 56–71, 2012.

Nguyen, V.-P.; Prins, C.; Prodhon, C. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. **European Journal of Operational Research**, v. 216, n. 1, p. 113–126, 2012.

Ostermeier, M.; Heimfarth, A.; Hübner, A. The multi-vehicle truck-and-robot routing problem for last-mile delivery. **European Journal of Operational Research**, v. 310, n. 2, p. 680–697, 2023.

Ouyang, Z.; Leung, E. K.; Huang, G. Q. Community logistics and dynamic community partitioning: A new approach for solving e-commerce last mile delivery. **European Journal of Operational Research**, v. 307, n. 1, p. 140–156, 2023.

Parada, L.; Legault, R.; Côté, J.-F.; Gendreau, M. A disaggregated integer L-shaped method for stochastic vehicle routing problems with monotonic recourse. **European Journal of Operational Research**, v. 318, n. 2, p. 520–533, 2024.

Pichka, K.; Bajgiran, A. H.; Petering, M. E. H.; Jang, J.; Yue, X. The two echelon open location routing problem: Mathematical model and hybrid heuristic. **Computers & Industrial Engineering**, v. 121, p. 97–112, 2018.

Prodhon, C.; Prins, C. A survey of recent research on location-routing problems. **European Journal of Operational Research**, v. 238, n. 1, p. 1–17, 2014.

Pureza, V.; Morabito, R.; Reimann, M. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the VRPTW. **European Journal of Operational Research**, v. 218, n. 3, p. 636–647, 2012.

Queiroga, E.; Sadykov, R.; Uchoa, E. A POPMUSIC matheuristic for the capacitated vehicle routing problem. **Computers & Operations Research**, v. 136, p. 105475, 2021.

Rahmani, Y.; Cherif-Khettaf, W. R.; Oulamara, A. The two-echelon multi-products location-routing problem with pickup and delivery: Formulation and heuristic approaches. **International Journal of Production Research**, v. 54, n. 4, p. 999–1019, 2016.

Reed, S.; Campbell, A. M.; Thomas, B. W. Does parking matter? The impact of parking time on last-mile delivery optimization. **Transportation Research Part E: Logistics and Transportation Review**, v. 181, p. 103391, 2024.

_____. The value of autonomous vehicles for last-mile deliveries in urban environments. **Management Science**, v. 68, n. 1, p. 280–299, 2021.

Schwengerer, M.; Pirkwieser, S.; Raidl, G. R. A variable neighborhood search approach for the two-echelon location-routing problem. In: EVOLUTIONARY Computation in Combinatorial Optimization. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. P. 13–24.

Senarclens de Grancy, G. An adaptive metaheuristic for vehicle routing problems with time windows and multiple service workers. **Journal of Universal Computer Science**, v. 21, n. 9, p. 1143–1167, 2015.

Senarclens de Grancy, G.; Reimann, M. Evaluating two new heuristics for constructing customer clusters in a VRPTW with multiple service workers. **Central European Journal of Operations Research**, v. 23, n. 2, p. 479–500, 2015.

_____. Vehicle routing problems with time windows and multiple service workers: A systematic comparison between ACO and GRASP. **Central European Journal of Operations Research**, v. 24, n. 1, p. 29–48, 2014.

Senna, F.; Coelho, L. C.; Morabito, R.; Munari, P. An exact method for a last-mile delivery routing problem with multiple deliverymen. **European Journal of Operational Research**, v. 317, n. 2, p. 550–562, 2024.

_____. **The two-echelon location-routing problem: A comparative analysis of novel and existing compact formulations**. [S.l.: s.n.], 2024. Technical report CIRRELT-2024-15, Université de Montréal.

Sluijk, N.; Florio, A. M.; Kinable, J.; Dellaert, N.; Van Woensel, T. Two-echelon vehicle routing problems: A literature review. **European Journal of Operational Research**, v. 304, n. 3, p. 865–886, 2023.

Solomon, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. **Operations Research**, INFORMS, v. 35, n. 2, p. 254–265, 1987.

Souza Neto, J. F.; Pureza, V. Modeling and solving a rich vehicle routing problem for the delivery of goods in urban areas. **Pesquisa Operacional**, v. 36, n. 3, p. 421–446, 2016.

Sutrisno, H.; Yang, C.-L. A two-echelon location routing problem with mobile satellites for last-mile delivery: Mathematical formulation and clustering-based heuristic method. **Annals of Operations Research**, v. 323, p. 203–228, 2023.

Tian, X.-D.; Hu, Z.-H. A branch-and-price method for a two-echelon location routing problem with recommended satellites. **Computers & Industrial Engineering**, v. 184, p. 109593, 2023.

Toth, P.; Vigo, D. **Vehicle routing: Problems, methods and applications**. 2. ed. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2014.

Wang, M.; Miao, L.; Zhang, C. A branch-and-price algorithm for a green location routing problem with multi-type charging infrastructure. **Transportation Research Part E: Logistics and Transportation Review**, v. 156, p. 102529, 2021.

Wang, Y.; Assogba, K.; Liu, Y.; Ma, X.; Xu, M.; Wang, Y. Two-echelon location-routing optimization with time windows based on customer clustering. **Expert Systems with Applications**, v. 104, p. 244–260, 2018.

Wang, Y.; Wang, X.; Wei, Y.; Sun, Y.; Fan, J.; Wang, H. Two-echelon multi-depot multi-period location-routing problem with pickup and delivery. **Computers & Industrial Engineering**, v. 182, p. 109385, 2023.

Wehbi, L.; Bektaş, T.; Iris, Ç. Optimising vehicle and on-foot porter routing in urban logistics. **Transportation Research Part D: Transport and Environment**, v. 109, p. 103371, 2022.

Yıldız, E. A.; Karaoğlan, İ.; Altiparmak, F. An exact algorithm for two-echelon location-routing problem with simultaneous pickup and delivery. **Expert Systems with Applications**, v. 231, p. 120598, 2023.

Zhao, Q.; Wang, W.; Souza, R. de. A heterogeneous fleet two-echelon capacitated location-routing model for joint delivery arising in city logistics. **International Journal of Production Research**, v. 56, n. 15, p. 5062–5080, 2017.