UNIVERSIDAD DE CONCEPCIÓN - CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows: A Perishable Good Case Study.

por:

Daniel Alfredo Neira González

Profesor Guía:

Doctor Carlos Contreras

Concepción, Enero de 2021

Tesis presentada a

DIRECCIÓN DE POSTGRADO
UNIVERSIDAD DE CONCEPCIÓN

Para optar al grado de

MAGISTER EN INGENIERÍA INDUSTRIAL

"Life can only be understood backwards;
but it must be lived forwards."

– Søren Kierkegaard.

"Be yourself;
everyone else is already taken."

– Oscar Wilde.

"Quia anima vivorum;
risus et mortuorum"

ABSTRACT

Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows: A Perishable Good Case Study.

Daniel Alfredo Neira González
January 2021

Thesis Supervisor: **Doctor Carlos Contreras**
Program: **Magíster en Ingeniería Industrial**

The Multi-Depot Multi-Trip Vehicle Routing Problem with Time Windows (MDMTVRPTW) stems from the problem faced by a bakery. First, it is necessary to analyze one of its components, namely, the vehicle routing problem with release dates. The vehicle routing problem with release dates (VRP-Rd) is a variant of the classic vehicle routing problem in which each customers order has a release date indicating the earliest time when the order is available at the depot for delivery. Hence, customers orders are loaded into vehicles after their release dates, denoting a quite common problem that arises in city logistics and last-mile distribution. In this work, we present a novel two-index compact formulation and lifted inequalities for VRP-Rd. The proposed formulation are compared, with and without lifted inequalities, against two existing compact formulations and a state-of-the-art algorithm reported in the literature, over a set of well-known benchmark instances. The results demonstrate that both variations of the proposed model outperform existing formulations, and they are competitive in terms of solutions quality with those obtained by the state-of-the-art algorithm. Additionally, a deep analysis is carried out to accelerate the optimization and improvement schemes are tested. Finally, it is worthy of highlighting that the resulting lifted formulation can be solved efficiently by commercial software without complicated algorithmic implementations.

Key Words: Vehicle Routing Problem, Release Dates, MIP

RESUMEN

Problema de ruteo de vehículos con múltiples depósitos, múltiples viajes y con ventanas de tiempo: Un caso de estudio para alimentos perecederos.

Daniel Alfredo Neira González
Enero 2021

Profesor Guía: **Doctor Carlos Contreras**
Programa: **Magíster en Ingeniería Industrial**

El problema de ruteo de vehículos con múltiples depósitos, múltiples viajes y con ventanas de tiempo (MDMTVRPTW, por sus siglas en inglés) nace del problema que enfrenta una panadería. Primero, fue necesario analizar uno de sus componentes, el problema de ruteo con tiempos de salida. El problema de ruteo con tiempos de salida (VRP-Rd, por sus siglas en inglés) es una variante del clásico problema de las rutas de vehículos en el que el pedido de cada cliente tiene una fecha de entrega que indica la hora más temprana en que este está disponible en el depósito para su entrega. Por lo tanto, los pedidos de los clientes se cargan en los vehículos después de sus fechas de liberación, lo que denota un problema bastante común que surge en la logística de la ciudad y en la distribución de última milla. En esta investigación, se presenta una novedosa formulación compacta de dos índices e inecuaciones ajustadas para el VRP-Rd. Utilizando un conjunto de instancias de referencias conocidas, se comparan la formulación propuesta, con y sin inecuaciones ajustadas, con dos formulaciones compactas existentes y un algoritmo del estado de larte reportado en la literatura. Los resultados demuestran que ambas variaciones propuestas superan a las formulaciones existentes, y son competitivas en términos de calidad de la solución con las obtenidas por el algoritmo del estado del arte. Adicionalmente, se lleva a cabo un profundo análisis para acelerar la optimización y esquemas de mejora. Por último, cabe destacar que la formulación con inecuaciones ajustadas resultante puede resolverse eficazmente utilizando *solvers* comerciales sin implementaciones algorítmicas complicadas.

Palabras Claves: Problema de Ruteo de Vehículos, Tiempos de Salida, MIP.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Objectives

1.1 Introduction

The vehicle routing problems (VRP) have been on the agenda for decades since their first appearance in the seminal work of Dantzig and Ramser, 1959. The classical VRP involves minimizing the total distance traveled by a fleet of vehicles to visit a set of costumers. Typically, VRPs assumes that vehicles depart from and return to a single depot, each customer demand is satisfied by a unique vehicle, and customer request are available at the depot for delivery at the beginning of the planning period. Variants and applications of the VRP are presented in Braekers, Ramaekers, and Van Nieuwenhuyse, 2016 and Mor and Maria Grazia Speranza, 2020.

VRPs and its variants have become especially important in the last two decades due to the rise of e-commerce. Only in the US, as of April 2020, there online searches in categories such as Books & Literature, Hobbies & Leisure, People & Society and Health have increased drastically to 187.67%, 119.83%, 106.90% and 96.47%, respectively. This surge in consumption pressures companies like Amazon, Ebay, Rakuten, Samsung, Walmart, Apple and Aliexpress to manage this online traffic (Andrienko, 2020; Augusta, 2020), which ultimately become additional sales (Dunn, Hood, and Driessen, 2020). Now, even though this dynamic environment has shaped the industry over the last few decades, it has been suddenly accelerated this year (Ivanov, 2020). Thereby, agile delivery and dynamic planning are paramount elements to maintain a competitive advantage.

Routing problems with release dates are extensions of the classical VRP in which each customer order has a release date indicating the earliest time that the order is available at the depot for delivery. The VRP assumes that customers' request are available at the depot for delivery at the beginning of the planning period. However, in many practical applications, this is not the case. These problems emerge in consolidation and distribution centers in city logistics. Goods to be delivered arrive over the planning period while vehicles are distributing other items available for delivery. The availability of new products for delivery forces the constant redrawing of delivery routes. Related problems arise in same-day delivery problems in e-commerce environment in which customer request arrive dynamically and must be delivered within the same day. Applications of routing problems with release dates include the pharmaceutical industry (Babagolzadeh et al., 2019), last mile delivery (Arslan et al., 2019), same-day delivery in the online retail industry (Voccia, Campbell, and Thomas, 2019) and food industry, adding value to online restaurants (Yildiz and M. Savelsbergh, 2019).

In this work, the vehicle routing problem with release dates (VRP-Rd) (Mor and Maria Grazia Speranza, 2020) is studied, which is defined as follows. Consider a direct graph $\mathcal{G} = (N, A)$, where $N = \{0, 1, ..., n\}$ is a set of nodes with 0 as the depot and n is set of customers, and $A = \{(i, j) \mid i, j \in N\}$ is a set of arcs. Each arc $(i, j) \in A$ is defined by a travel time t_{ij} and a distance d_{ij}. A fleet of R homogeneous vehicles with capacity Q is located at the depot, each performing at most one route, and each customer $i \in N$ has a request of q_i units of a given product available for delivery at the depot after time r_i (release date). The goal is to create a set of routes that minimize the total traveled time so that each vehicle starts and ends at the depot, the demand of each customer is satisfied, each customer order is loaded into vehicles after its release dates, and the vehicle capacity is never exceeded.

A new two-index compact formulations is presented and valid inequalities for VRP-Rd are proposed. Results are compared with an existing three-index formulation (L. Liu, Kunpeng Li, and Z. Liu, 2017), and an adapted two-index mathematical formulation reported for a variant of the VRP-Rd (W. Li et al., 2020). Moreover, formulations are tested against algorithms proposed

(L. Liu, Kunpeng Li, and Z. Liu, 2017) for the VRP-Rd in terms of quality. The proposed formulation is tested over a set of 27 instances of the CVRPLib benchmark instances adapted to VRP-Rd, containing up to 80 customers and up to 10 vehicles. Finally, we perform a sensitivity analysis, by varying weights on the objective function; and we prove different improvements schemes to obtain better outcomes.

In general, our results suggest that: i) our proposed models are effective and outperform both the three and two-index formulation reported in the literature for the VRP-Rd, (ii) our lifted formulation of the VRP-Rd is competitive (in terms of solution quality) when compared to the heuristic proposed in ibid. on instances with up to 80 customers, (iii) our vanilla model is more suited when the number of nodes is less than 56, then both of our formulations performs similarly; (iv) our models outperform the existing one and the vanilla is able to find optimal solutions when weighted in favor of the construction of circuits; and (v) when different improvement schemes are tested, our vanilla formulation with an Initial Solution insertion and a guided local search yields better solution and mean gaps, but the vanilla formulation version without Improvement tends to obtain more best objective functions.

The main contribution of this research is providing a two-index compact model for the VRP-Rd, which can be adapted for its extensions. Additionally, valid inequalities are presented, lifted constraints that help prune the solution space and easy-to-follow improvement schemes. Moreover, the proposed models are effective and can be used on any commercial solver without requiring problem dependent implementations such column generation or cumbersome heuristic procedures, which is a great contribution for practitioners.

1.2 Objectives

General Objectives

Study the problem of Vehicle Routing with release dates (VRP-Rd) from the perspective of mathematical programming and propose a compact formulation for the problem.

Specific Objectives

- Generate mathematical models for the VRP-Rd problem, then program and test them in `Gurobi` using the `Python` programming language.
- Propose a solution method for VRP-Rd. Study its properties, such as response efficiency in comparison with other possible models for the problem.
- Develop valid inequalities specific to the problem.
- Implement algorithms that generate initial solutions.
- Implement algorithms proposed in the literature to obtain points of comparison.
- Propose, characterize and analyze possible extensions for the VRP-Rd problem.

1.3 Outline

The remainder of this thesis is structured as follows. First, Chapter 2 presents a brief review of VRP-Rd and existing formulations for this problem. Next, Chapter 3 presents mathematical models for VRP-Rd and valid inequalities. Then, Chapter 4 reports computational results when solving a large-set of benchmarks instances reported in the literature. Finally, this work concludes and proposes possible further researches in Chapter 5. Additionally, some demonstration can be found in Appendix A

Chapter 2

Literature Review

Release dates have been widely studied in scheduling literature (Carlier, 1987; Kai Li and Cheng, 2010). However, in the routing literature, release dates have received less attention. These routing problems can be identified in two different areas: i) in city logistic problems (Archetti, Feillet, Mor, and M Grazia Speranza, 2018; B. C. Shelbourne, Battarra, and Potts, 2017) and, ii) dynamic vehicle routing problems (Archetti, Feillet, Mor, and Maria Grazia Speranza, 2020; Darvish, Coelho, and Laporte, 2020; Klapp, Erera, and Toriello, 2018a,b; Reyes, Erera, and M. W. Savelsbergh, 2018; Yildiz and M. Savelsbergh, 2019).

2.1 City logistics problems with release dates

City logistics models intend to minimize the inconveniences associated with the transport of goods in urban areas while promoting their economic and social development, which involves the movement coordination of different kind of trucks between shippers and carriers (Crainic, Ricciardi, and Storchi, 2009). These requirements impose different types of constraints, such as time-dependency, multi-level and multi-trip organization of the distribution, dynamic information, among others. For a recent review in city logistics, see Cattaruzza, Absi, Feillet, and González-Feliu, 2017. City logistics models reported in the literature involving release dates include the traveling salesman problem (TSP), classical VRP (CVRP), multi-period VRP

(MPVRP), VRP with order picking and multi-trip VRP.

The TSP with release dates has been studied in Archetti, Feillet, Mor, and M Grazia Speranza, 2018; Archetti, Feillet, and M Grazia Speranza, 2015. Archetti, Feillet, and M Grazia Speranza, 2015 demonstrates that for underlying network structures, the problem can be solved in polynomial time. They study the uncapacitated VRP-Rd under two possible conditions: i) when a limit time for completing the distribution is set, and the total distance is the function to be minimized, and ii) when no delivery limits are set and delivery completion time is minimized. Archetti, Feillet, Mor, and M Grazia Speranza, 2018 seeks to minimize the completion time of the uncapacitated TSP with release dates. Additionally, they assume that the vehicles can perform multiple routes with the aim to minimize the total time needed to serve all customers. The proposes MIP formulation and iterated local search to solve the problem.

Variants of the CVRP with release dates are also reported in the literature. L. Liu, Kunpeng Li, and Z. Liu, 2017 introduces a variant of the CVRP, in which release time is considered (problem observed in the e-commerce industry), and proposes a tabu search algorithm (TS). B. Shelbourne, 2016 and B. C. Shelbourne, Battarra, and Potts, 2017 extend the vehicle routing and scheduling problem, by including a release date to each customer and a due date to each order. They evaluate MIP formulations, efficient heuristics, and a Dantzig-Wolfe decomposition. Finally, Jaikishan and Patil, 2019 studies the VRP with release dates and due dates, and presents reactive GRASP heuristics to tackle the problem.

The multi-period VRP (MPVRP) with release dates is studied in Archetti, Jabali, and M Grazia Speranza, 2015. They consider the case in which a release and a due date characterize each customer, that must be served at the first and last period, respectively. They develop several four-index formulations, which are solved with a branch-and-cut algorithm.

VRP with release dates have also been addressed when considering routing jointly with order picking (OPVRP). Schubert, Scholz, and Wäscher, 2018 studies the order assignment and sequencing, and the vehicle routing problem with due dates of a supermarket chain where deliveries for supermarkets are picked in a central warehouse. An iterated local search algorithm solves the two subproblems simultaneously. Then Moons et al., 2018 integrates the order picking and vehicle routing in a B2C e-commerce context, and solve this problem by a mixed-integer linear programming formulation. Their results show how considering the two problems together results in an average cost saving of 14%.

The VRP assumes that a vehicle can perform at most one trip or route during the routing planning horizon; however, this is not the case in practical situations. The multi-trip VRP with release dates (MTVRP-rd) assumes that vehicles can perform multiple trips and orders have release dates. Cattaruzza, Absi, and Feillet, 2016 considers the MTVRP-rd in the context of delivery systems involving city distribution centers. They solve it using a population-based metaheuristic algorithm. Campelo et al., 2019 tackles a VRP that considers customers with multiple daily deliveries, time windows and release dates, which is solved using an instance size reduction algorithm and a mathematical-programming-based decomposition. Finally, W. Li et al., 2020 revisits the problem studied in Cattaruzza, Absi, and Feillet, 2016, but they focus the problem on last-mile delivery for e-commerce. They formulate a MILP and solve it using an adaptive large neighborhood search algorithm combined with a labelling procedure (ALNS-L).

Table 2.1 presents a summary of the main city logistics models reported in the literature. The first column denotes the authors, while columns 2-4 indicate the version studied: TSP, CVRP, MPVRP, OPVRP and MTVRP . Columns 5-7 display the type of mathematical formulation reported (if any reported). To the best of our knowledge, a two-index formulation for VRP-Rd has not been reported in the literature. However, W. Li et al., 2020 have developed a two-index formulation for the MTVRP with release dates, based on a non-overlapping trip packing of customers to vehicles, which we will adapt to VRP-Rd and then use it to compare our results in Chapter 4.

2.2 Dynamic vehicle routing problems with release dates

The dynamic vehicle routing problem (DVRP) can be defined as variants of the VRP where: i) Not all important information is known at the beginning of the route planning and ii) the relevant information may change throughout the routing process (Psaraftis, Wen, and Kontovas, 2016).

Table 2.1: Summary of City logistics literature.

	Problem					Mathematical formulation		
References	TSP	CVRP	MPVRP	OPVRP	MTVRP	4-index	3-index	2-index
Archetti, Feillet, and M Grazia Speranza, 2015	✓							
Archetti, Jabali, and M Grazia Speranza, 2015			✓			✓		
Cattaruzza, Absi, and Feillet, 2016					✓			
B. C. Shelbourne, Battarra, and Potts, 2017			✓				✓	
L. Liu, Kunpeng Li, and Z. Liu, 2017		✓					✓	
Schubert, Scholz, and Wäscher, 2018				✓		✓		
Moons et al., 2018				✓			✓	
Archetti, Feillet, Mor, and M Grazia Speranza, 2018	✓					✓		
Jaikishan and Patil, 2019		✓				✓		
Campelo et al., 2019					✓	✓		
W. Li et al., 2020					✓			✓
Our Models		✓						✓

One of the common characteristics found in this type of problems is the difficulty of forecasting the precise moment when orders or goods are ready for delivery; otherwise, the company incurs in unnecessary holding costs and loss of quality. Hence, knowing the release time is key when managing delivery. Among DVRP problems, release time can be found on same-day delivery systems (Klapp, Erera, and Toriello, 2018a,b; Ulmer, Thomas, Campbell, et al., 2020; Ulmer, Thomas, and Mattfeld, 2019; Voccia, Campbell, and Thomas, 2019) and other applications.

Same-day delivery systems (SDD), can be described as a problem in which delivery requests arrive dynamically throughout a service day and must be delivered on the same day. Klapp, Erera, and Toriello, 2018b presents two variants of this problem: i) a deterministic case, solved using dynamic programming to determine an *a-priori* policy for predetermined routes, and ii) fully dynamic policies, for which the authors proposed a heuristic procedure and compute dual bounds. Klapp, Erera, and Toriello, 2018a extends their previous work by formulating the Dynamic Dispatch Waves Problem (DDWP), where at each decision wave (epoch), the systems operator decides whether or not to dispatch a single-vehicle loaded with orders ready for service in order to minimize vehicle travel costs and penalties for unnerved requests. Reyes, Erera, and M. W. Savelsbergh, 2018 proposes a generalization of the work presented in Archetti, Feillet, and M Grazia Speranza, 2015, accomplishing polynomial time complexity proofs for the same-day delivery optimization problem, testing both single and multiple vehicle variations, and assuming that customers are located on a half-line. Furthermore, Voccia, Campbell, and Thomas, 2019 extends the study to a multi-vehicle dynamic pickup and delivery problem with time constraints associated with same-day delivery logistics. Ulmer, Thomas, and Mattfeld, 2019 considers an SDD where vehicles are allowed to return to the depot before having completed their distribution to load the parcels of new customers. A probability distribution characterizes unknown customers. They solve this problem by utilizing approximate dynamic programming. Ulmer, Thomas, Campbell, et al., 2020 combines the dynamic pickup and delivery problem (DPD) and SDD with characteristics such as postponement, scalability and anticipation; in context of restaurant meal delivery, similar to today's mobile applications. The probability distributions on the time and location of meal requests are known. Before the delivery, the selected vehicle has to pick up the meal at the restaurant. The meal preparation time is random, and therefore, the vehicle may wait at the restaurant to pick up the meal. To address these challenges, they present an anticipatory customer assignment (ACA) policy, which deals with stochastic by postponing the assignment decisions for selected customers, allowing flexible assignments.

Others DVRP with release dates has also been reported in the literature. Archetti, Feillet, Mor, and Maria Grazia Speranza, 2020 adresses the dynamic traveling salesman problem with stochastic release dates (DTSP-srd). In the DTSP-srd release dates are stochastic and dynamically updated as distribution occurs. The goal is to minimize the total time needed to serve all customers, which is the sum of the traveling time and the waiting time at the depot. They represent the problem as a Markov decision process and solve it using a reoptimization approach comprised of two models. The first is a stochastic model that estimates release dates, where the second is deterministic and uses the release date estimations as an input. Yildiz and M. Savelsbergh, 2019 presents the meal delivery routing problem (MDRP) in last-mile logistics. An order is expected to be delivered as quickly as possible and within minutes of the food becoming ready. They introduce a formulation for the MDRP, where perfect information about order arrivals is assumed. Then, they develop a simultaneous column- and row-generation method. Finally, Darvish, Coelho, and Laporte, 2020 tackles a variant of the multi-period routing problem in which deliveries may occur between release and due dates. The release date of each product is stochastic, and customer orders arrive dynamically over a planning horizon. They provide a model that at the beginning of the planning horizon, solves the planning of incoming orders. Then, when a distribution interruption occurs, a new resource model is solved in an iterative way over the remaining time. Both models are solved by means of a branch-and-cut.

Table 2.2 presents a summary of the main DVRP extensions involving release dates reported in the literature. The first column contains the authors, while columns 2-5 indicate the extensions studied: SDD (same-day delivery), DTSP (dynamic TSP), MDRP (meal delivery problem), and others. Finally, columns 6-8 indicates the type of compact formulation reported (if any reported). As far as we know, a two-index formulation for any DVRP with release dates (related to VRP-Rd) has not been reported.

Table 2.2: Summary of DVRP literature.

	Problem				Mathematical formulation		
References	SDD	DTSP	MDRP	Other	4-index	3-index	2-index
Klapp, Erera, and Toriello, 2018b	✓						
Klapp, Erera, and Toriello, 2018a	✓					✓	
Reyes, Erera, and M. W. Savelsbergh, 2018	✓						
Voccia, Campbell, and Thomas, 2019	✓						
Ulmer, Thomas, and Mattfeld, 2019	✓						
Ulmer, Thomas, Campbell, et al., 2020	✓						
Yildiz and M. Savelsbergh, 2019			✓				
Darvish, Coelho, and Laporte, 2020				✓	✓		
Archetti, Feillet, Mor, and Maria Grazia Speranza, 2020		✓				✓	

Chapter 3

Mathematical formulation

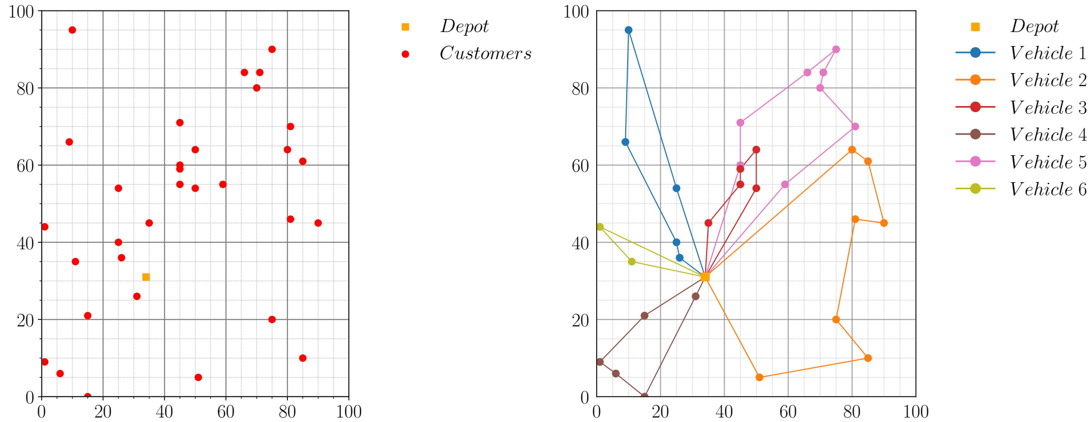
The objective of the VRP-Rd is to plan a set of vehicle journeys minimizing the total traveled time by a fleet subject to the following constraints:

1. Each vehicle journey starts and ends at the depot;

2. Each customer's request is loaded into vehicles after r_i and delivered;

3. In each journey, the vehicle capacity (Q) cannot be exceed.

To help the reader to understand our formulation, we present a solution to an instance of VRP-Rd. We use the problem `A-n33-k6` from the CVRP benchmark consisting of 33 nodes (one depot and 32 customers), and a fleet of six homogeneous vehicles. The locations of the depot and customers are depicted in Figure 3.1(a), while a feasible solution for this problem is shown in Figure 3.1(b) in which six routes, one for each vehicle, have been created. Figure 3.2 display the impact of release dates on the vehicles departing time from depots. For each vehicle, the x-axis display time and the y-axis depicts the nodes visited by the vehicle, from top to bottom, with v_0 being the depot and v_1 - v_{32} the customers. Besides, the customer's release dates are shown as a bracket and the departing time for each vehicle from the depot is marked by a vertical gray dotted line. This last figure helps to visualize how the vehicles starts their routes after the largest release date on the visited customers.



(a) Instance A-n33-k6

(b) Feasible solution for A-n33-k6

Figure 3.1: Example of problem instance and solution A-n33-k6.

3.1 Mathematical formulation

This work presents a two-index MIP formulation for VRP-Rd. Let us first introduce the following notation. Let $D = \{0, ..., R-1\}$ be the set of depot copies representing the starting point of R vehicles. We make a copy of the depot for each vehicle so each their departing times from the depot can be captured. A last copy of the depot determines the final return, which is shared for all vehicles and is defined as $f = R$. Additionally, let $C = \{R+1, ..., R+n\}$ be the set of customers, $\mathcal{N} = \{D \cup \{f\} \cup C\}$ be the complete set of nodes.

The decision variables are as follows. x_{ij} is binary variable equal to 1 if i precedes j and 0, otherwise, $\forall i, j \in \mathcal{N}$; y_{ij} represent the flow of product in arc (i, j), $\forall i, j \in \mathcal{N}$; u_i is a continuous variable capturing the arrival time to node i (relative to the start of the journey), $i \in \mathcal{N}$; and h_i is a continuous variable capturing the journey starting time that visit node i, $\forall i \in C$.

$$\textbf{VRP-Rd:} \quad \textbf{minimize} \sum_{(i,j)\in A, i\neq j} t_{ij}x_{ij} + \sum_{d\in D} u_d \tag{3.1}$$

subject to:

$$\text{(Assignment constraints)} \quad \sum_{j\in C} x_{dj} = 1, \quad \forall d \in D \tag{3.2}$$

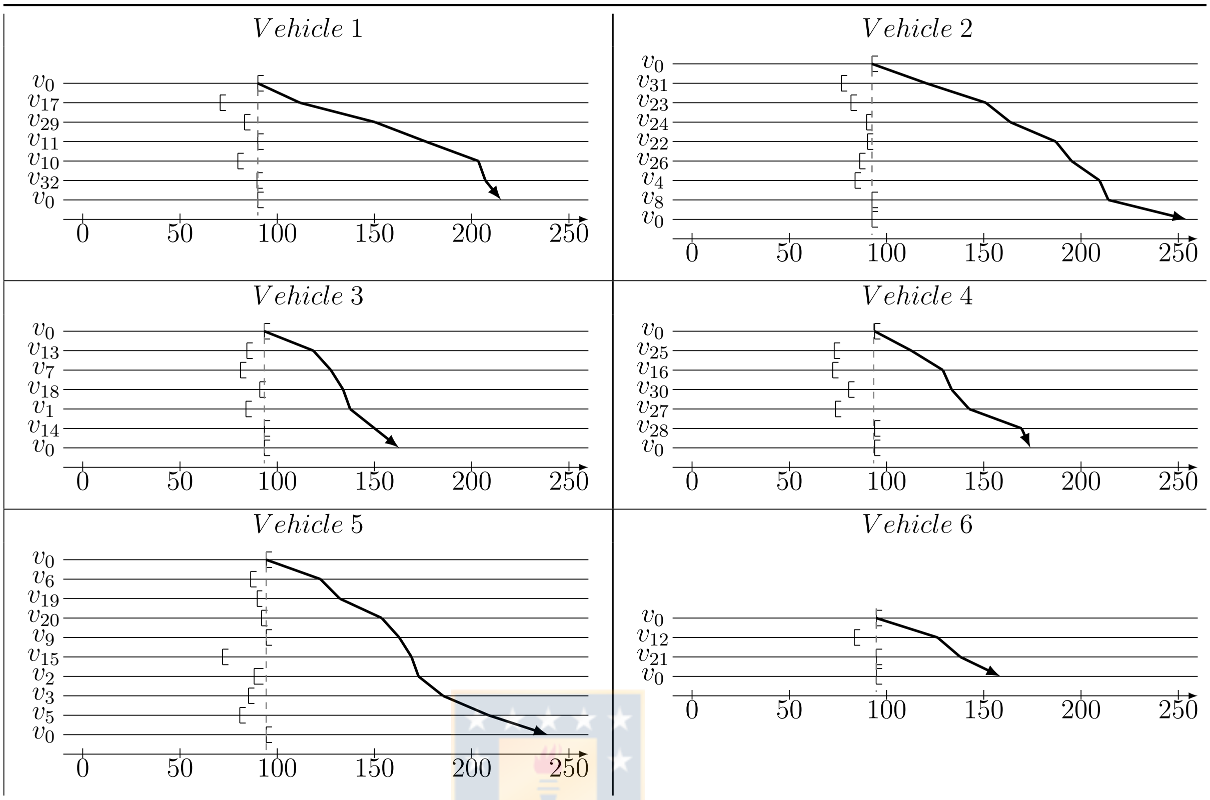


Figure 3.2: Packing of customer in VRP-Rd.

$$\sum_{j \in C} x_{jf} = R \tag{3.3}$$

$$\sum_{j \in C \cup \{f\}, i \neq j} x_{ij} = 1, \ \ \forall i \in C \tag{3.4}$$

$$\sum_{j \in D \cup C, i \neq j} x_{ji} = 1, \ \ \forall i \in C \tag{3.5}$$

(Demand & capacity constraints) $$y_{ij} \leq Q x_{ij}, \ \ \forall (i,j) \in A, i \neq j \tag{3.6}$$

$$\sum_{j \in C \cup \{f\}, j \neq i} y_{ij} - \sum_{j \in D \cup C, j \neq i} y_{ji} = d_i, \ \ \forall i \in C \tag{3.7}$$

(Journey duration constraints) $$u_i + t_{ij} - u_j \leq M_1(1 - x_{ij}), \ \ \forall (i,j) \in A, \ \ i \neq j \tag{3.8}$$

(Release date constraints) $$u_d \geq h_i - M_2(1 - x_{di}), \ \ \forall d \in D, \ \ \forall i \in C \tag{3.9}$$

$$h_j \geq h_i - M_3(1 - x_{ij}), \ \ \forall i, \ j \in C, \ \ i \neq j \tag{3.10}$$

$$h_i \geq h_j - M_4(1 - x_{ij}), \ \ \forall i, \ j \in C, \ \ i \neq j \tag{3.11}$$

$$h_i \geq r_i, \ \ \forall i \in C \tag{3.12}$$

$$u_d \leq u_{d+1}, \ \ \forall r \in D \quad \text{(Symmetry breaking constraints)} \tag{3.13}$$

$$x_{ij} \in \{0,1\}, \ \ \forall i,j \in \mathcal{N}, \ i \neq j \quad \text{(Variable nature constraints)} \tag{3.14}$$

$$y_{ij} \geq 0, \ \ \forall i,j \in \mathcal{N}, \ i \neq j \tag{3.15}$$

$$u_i \geq 0, \ \ \forall i \in \mathcal{N} \tag{3.16}$$

$$h_i \geq 0, \ \ \forall i \in C. \tag{3.17}$$

The objective function (3.1) minimizes the total traveled time by all vehicles and the release time of each vehicle. Constraints (3.2) to (3.5) are typical VRP assignment constraints guaranteeing vehicle flow conservation. Constraints (3.2) and (3.3) force that each vehicle departs from its depot and returns to the final depot exactly once. Constraints (3.4) and (3.5) force that each customer is visited exactly once. Constraints (3.6) to (3.7) meet customer demand requirements and satisfy vehicle maximum capacity. Due to Constraints (3.6) only arcs that are in used can carry flows of goods. The arrival times are calculated using constraints (3.8), which also prohibit subtours. Constraints (3.9) to (3.12) impose the release date on customer's requests. Constraints (3.9) compute the trip's starting time which has to be greater than h_i if i is the first customer visited from the depot, while constraints (3.10) and (3.11) make the trip's starting time identical to all customer on a given trip. Constraints (3.12) impose that the trip's starting time must be greater than the customer's release dates. Constraints (3.13) help to break symmetries forcing journey's starting times to be ordered from lowest to highest Finally, Constraints (3.14) to (3.17) set the nature of all decision variables.

In Table 3.1[1], for each constraint, we present the structural parameter value (or big-M parameter) derived from the problem parameters (See A.1 for a formal mathematical derivation).

3.2 Lifting some constraints for VRP-Rd

In the same fashion followed on in Desrochers and Laporte, 1991 and corrected in Kara, Laporte, and Bektas, 2004, we propose a lifted restriction for constraints (3.8) as follows:

[1] The value $\max u_f$ can be set to $|N| \max\limits_{i,j \in C, i \neq j} t_{ij}$, a further research is undergone in Apendix A.2

Table 3.1: Structural parameters for parameters for our Formulations

Constraints	*Group*	*M value*
(3.8)	Journeys duration constraints	$M_1 = \max u_f + \max_{i,j\in C, i\neq j} t_{ij} - \min_{i\in C} r_i - \min_{d\in D, j\in C} t_{dj}$
(3.9)	Release date constraints	$M_2 = \max_{i\in C} r_i - \min_{i\in C} r_i$
(3.10)	Release date constraints	$M_3 = \max_{i\in C} r_i - \min_{i\in C} r_i$
(3.11)	Release date constraints	$M_4 = \max_{i\in C} r_i - \min_{i\in C} r_i$

Proposition 3.2.1 *The constraints*

$$u_i + t_{ij} - u_j + (M_1 - t_{ij} + \min_{j\in C} r_j + min_{d\in D, j\in C} t_{dj} - \max u_f + \min_{i\in C} t_{if})\, x_{ji} \leq M_1(1 - x_{ij}), \;\; \forall i, j \in C, \;\; i \neq j \tag{3.18}$$

are valid inequalities for the VRP-Rd.

Proof 1 *Consider the following general constraints:*

$$u_i + t_{ij} - u_j + (M_1 - t_{ij} + \alpha_{ji})\, x_{ji} \leq M_1(1 - x_{ij}), \;\; \forall i, j \in C, \;\; i \neq j \tag{3.19}$$

Where we want to find the largest value for the given α_{ji}, being constraints (3.19) *valid. It can be seen that when $x_{ji} = 0$:*

$$u_i + t_{ij} - u_j \leq M_1(1 - x_{ij}), \;\; \forall i, j \in C, \;\; i \neq j \tag{3.20}$$

it holds regardless of the value of α_{ji}. Whereas, when $x_{ji} = 1$ (customer i is visited right after visiting customer j, then $x_{ij} = 0$, constraints (3.19) *become:*

$$\begin{aligned} u_i + t_{ij} - u_j + M_1 - t_{ij} + \alpha_{ji} &\leq M_1 \\ \implies u_i - u_j + \alpha_{ji} &\leq 0 \\ \implies \alpha_{ji} \leq u_j - u_i, \;\; \forall i, j \in C, \;\; i &\neq j \end{aligned} \tag{3.21}$$

Thus, to find the largest valid value for α_{ji}, the lesser value u_j for node j will be met when the *current vehicle have leaved the depot, therefore,* $x_{dj} = 1$ *and from constraints* (3.9)*, we have* $u_d \geq h_j$*, and given* (3.12) $\implies u_d \geq r_j \geq \min_{j \in C} r_j$*, considering* (3.8) $\implies u_d + t_{dj} \leq u_j$*; then* $u_j \geq \min_{j \in C} r_j + min_{d \in D, j \in C} t_{dj}$*. On the other hand, the biggest condition for node* i *will be met when it precedes the return to the depot, therefore,* $x_{if} = 1$*, then given* (3.8) $\implies u_i + t_{if} \leq u_f \implies u_i \leq u_f - t_{if}$ *which could be tightened* $\implies u_i \leq \max u_f - \min_{i \in C} t_{if}$*, where how to calculate* $\max u_f$ *is further discuss in Appendix A.2. Then the largest value for* α*, in order to constraints* (3.21) *be valid, can be calculated as:*

$$\alpha_{ji} = \min_{j \in C} r_j + min_{d \in D, j \in C} t_{dj} - (\max u_f - \min_{i \in C} t_{if}), \ \ \forall i, j \in C, \ \ i \neq j \tag{3.22}$$

Proposition 3.2.2 *The constraints*

$$u_i + t_{ij} - u_j + (M_1 - t_{ij} - t_{ji})\, x_{ji} \leq M_1(1 - x_{ij}), \ \ \forall i, j \in C, \ \ i \neq j \tag{3.23}$$

are super valid inequalities (Israeli and Wood, 2002) for the VRP-Rd.

Proof 2 *Consider the following general constraints: We seek to determine the largest value for the given* α_{ji} *such that constraints* (3.19) *are valid. It can be seen that when* $x_{ji} = 0$*: it holds regardless of the value of* α_{ji}*. Whereas, when* $x_{ji} = 1$ *(customer* i *is visited right after visiting customer* j*, then* $x_{ij} = 0$*, constraints* (3.19) *become: From constraints* (3.8) *for* $x_{ji} = 1$*, we obtain:* $u_j - u_i \leq -t_{ij}$
Hence, $\alpha_{ji} \leq u_j - u_i \leq -t_{ij}$*, and we set*

$$\alpha_{ji} = -t_{ij}$$

Proposition 3.2.3 *The constraints*

$$h_j - M_3\, x_{ji} \geq h_i - M_3(1 - x_{ij}), \ \ \forall i, \ j \in C, \ \ i \neq j \tag{3.24}$$

$$h_i - M_4\, x_{ji} \geq h_j - M_4(1 - x_{ij}), \ \ \forall i, \ j \in C, \ \ i \neq j \tag{3.25}$$

are valid inequalities for the VRP-Rd.

Proof 3 *Consider the following general constraints:*

$$h_j - (M_3 + \alpha_{ji})\, x_{ji} \geq h_i - M_3(1 - x_{ij}),\ \ \forall i,\ j \in C,\ \ i \neq j$$

We seek a valid for α_{ji} such that constraints (3.26) *are valid. Two cases may be considered. In the first case, $x_{ji} = 0 \implies h_j \geq h_i - M_3(1 - x_{ij}),\ \ \forall i,\ j \in C,\ \ i \neq j$, which holds regardless of the value of α_{ji}.*
In the second case, $x_{ji} = 1$ and thus $x_{ij} = 0$, we obtain:

$$\begin{aligned} h_j - M_3 + \alpha_{ji} &\geq h_i - M_3 \\ \implies h_j + \alpha_{ji} \geq h_i \implies \alpha_{ji} &\geq h_i - h_j,\ \ \forall i,\ j \in C,\ \ i \neq j \end{aligned}$$

And because, as the problem VRP-Rd is defined, when two customers, in this case j and i, are in the same journey, both take the same value of h that defines the release time of the current vehicle, and therefore the shortest value that takes $\alpha_{ji} = 0$. Thus we obtained constraint (3.24)*.*

Constraint (3.25) can be demonstrated following the same ideas.
Additionally, we lift the capacity constraints (3.6) as follows (Gavish and Graves, 1978):

$$y_{ij} \leq (Q - q_j)\, x_{ij},\ \ \forall (i, j) \in A, i \neq j \tag{3.26}$$

and add the following constraints:

$$y_{ij} \geq q_i x_{ij},\ \ \forall (i, j) \in A, i \neq j \tag{3.27}$$

Chapter 4

Computational Experiments

In this section, we present the results obtained by the proposed formulation in different benchmarks instances. All models were programmed in `python` and solved via `Gurobi version 9`, with default parameters. Additionally, the runs were distributed to a cluster of 30 computer nodes, each with an `Intel(R) Xeon(R) CPU E3-1270 v6 @ 3.80GHz` and `64 GB of RAM` running `Ubuntu 18.04.2 LTS`. Moreover, we set a time limit of three hours (10,800 seconds) per instance and allowed each instance to be optimized using `8 Gbs of RAM`.

We test the following models:

Model 1: Our original model consisting of constraints (3.1)- (3.17).

Model 2: Our lifted model, which is generated by changing constraints (3.6) for constraints (3.26); replacing in our original model constraints (3.8), (3.10) and (3.11) for their lifted version (3.23), (3.24) and (3.25) , respectively; and adding constraints (3.27).

4.1 Instances

We use the modified VRP-Rd benchmark instances used in L. Liu, Kunpeng Li, and Z. Liu, 2017. These instances are derived from the well-known CVRP benchmark[1] and modified as presented in the paper mentioned above. The dataset provides three series A, B and P, which adds up to 72 instances. We solve series A consisting of 27 instances with a range of nodes from 23 to 80 having from 5 to 10 vehicles. The instances are identified by the notation A-n32-k5, where the first, second, and third terms indicate, the series of the chosen instance, the number of nodes, and the available number of vehicles, respectively. Additionally, for each customer in each instance, the release times are generated randomly from the range $[1/8, 1/4] * T/R$, where T represents the travel time of the grand route which visits all customers from index 1 to n, and R is the number of vehicles.

4.2 Comparison with existing formulations

We start by comparing our MIP models (Modesl 1 and 2) against the ones reported in ibid. (three-index formulation) and W. Li et al., 2020 (two-index formulation), the latter modified to our problem. We present results obtained for the 27 instances explained above in Table 4.1. The first column indicates the instance name; and then, we display for each formulation: the best integer solution found (Z_{ip}); the required time to solve the instance (T); and the optimality gap reported by `Gurobi` (Gap). We report "Not" in the Gap column when the model is unable to obtain a feasible solution. For each instance, we highlight in bold the minimum values in Z_{ip} and Gap. We present in the last two rows a summary of the results obtained. "Mean" denotes the average value for each column; while "Best" indicates the number of times that the formulation obtains minimum Z_{ip} value among all formulations.

First, we compare the results obtained by the models described in L. Liu, Kunpeng Li, and Z. Liu, 2017 and W. Li et al., 2020 (see Table 4.1). The latter outperforms the former by reaching

[1]CVRP benchmark instances are available on the website `https://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/`

the minimum Z_{ip} value in 24 out of 27 instances. However, despite this behaviour, the big drawback of the model reported in W. Li et al., 2020 when compared to L. Liu, Kunpeng Li, and Z. Liu, 2017 is in finding lower bounds (Gap), which obtain minimum Gap values in only one instance. Hence, we will use the two-index formulation reported in W. Li et al., 2020 for comparison with the proposed formulations.

Now, we analyze the performance of our first proposed formulation, Model 1, with the two-index formulation reported in ibid. Model 1 obtain minimum Zip value in 25 out of 27 instances and reaches an average Gap of 15.80%, while the model presented in ibid. obtain an average Gap of 52.86%. Hence, Model 1 outperforms the formulations reported in L. Liu, Kunpeng Li, and Z. Liu, 2017 and W. Li et al., 2020; however, it is unable to obtain a feasible solution to instance `A-n61-k9`.

Knowing that Model 1 outperforms the existing formulations, we compare it with our improved formulation, Model 2, including lifted constraints derived in Chapter 3. Model 2 surpasses Model 1 by obtaining minimum Z_{ip} value in 17 out of 27 instances, and by reaching feasible solutions to all cases. The average Gap on the instances solved by both formulations by Models 1 and 2 are 15.80% and 15.64%, respectively. Thus, Model 2 dominates both Model 1 and all formations reported in the literature.

Table 4.1: Results on VRP-Rd for our both models vs formulation from L. Liu, Kunpeng Li, and Z. Liu, 2017 and W. Li et al., 2020

Instance	Model 1			Model 2			Three-index formulation (L. Liu, Kunpeng Li, and Z. Liu, 2017)			Two-index formulation (W. Li et al., 2020)		
	Z_{ip}	T	Gap	Z_ip	T	Gap	Z_ip	T	Gap	Z_ip	T	Gap
A-n32-k5	**1,214.00**	10,800.03	8.13%	1,214.72	10,800.04	**7.64%**	1,325.70	10,800.02	24.85%	1,264.69	10,800.01	17.17%
A-n33-k5	**1,039.73**	10,800.02	**7.43%**	**1,039.73**	10,800.04	8.29%	1,083.04	10,800.01	20.00%	1,110.06	10,800.01	27.05%
A-n33-k6	**1,048.93**	10,800.01	8.89%	**1,048.93**	10,800.01	**8.51%**	1,050.49	10,800.01	19.54%	1,147.09	10,800.04	30.57%
A-n34-k5	1,208.11	10,800.02	10.71%	**1,202.10**	10,800.02	**10.15%**	1,309.42	10,800.03	24.07%	1,255.51	10,800.01	31.08%
A-n36-k5	1,207.77	10,800.01	12.20%	**1,207.14**	10,800.00	**11.22%**	1,380.08	10,800.01	33.02%	1,299.04	10,800.01	34.33%
A-n37-k5	**998.84**	10,800.02	**7.33%**	1015.96	10,800.02	9.53%	1,178.23	10,800.03	26.89%	1,139.20	10,800.01	40.24%
A-n37-k6	1,348.42	10,800.02	11.55%	**1341.54**	10,800.01	**11.10%**	1,547.33	10,800.02	37.47%	1,438.00	10,800.01	39.21%
A-n38-k5	**1,218.49**	10,800.01	**11.65%**	1,219.88	10,800.02	11.68%	1,354.36	10,800.03	28.76%	1,320.15	10,800.01	30.76%
A-n39-k5	1,307.44	10,800.01	14.19%	**1300.48**	10,800.01	**14.09%**	1,482.35	10,800.03	30.96%	1,407.18	10,800.07	39.17%
A-n39-k6	**1,336.60**	10,800.02	11.25%	1338.90	10,800.01	**10.54%**	1,542.81	10,800.04	32.17%	1,532.25	10,800.02	39.11%
A-n44-k7	**1,482.66**	10,800.01	**14.69%**	1,505.66	10,800.01	15.82%	1,723.00	10,800.02	31.27%	1,698.66	10,800.01	52.94%
A-n45-k6	1,572.27	10,800.02	**18.04%**	**1,548.28**	10,800.03	18.15%	2,078.20	10,800.01	41.35%	1,938.00	10,800.02	56.97%
A-n45-k7	1,705.44	10,800.02	**14.33%**	**1,694.86**	10,800.03	14.77%	2,001.39	10,800.02	39.81%	1,858.43	10,800.01	51.06%
A-n46-k7	**1,453.64**	10,800.02	10.27%	1,454.30	10,800.00	**10.13%**	1,831.86	10,800.03	36.82%	1,768.81	10,800.03	50.38%
A-n48-k7	1,683.34	10,800.02	17.91%	**1,672.80**	10,800.02	**17.43%**	1,975.15	10,800.07	41.32%	1,933.18	10,800.01	56.44%
A-n53-k7	1,685.39	10,800.01	18.01%	**1,675.88**	10,800.03	**17.18%**	2,408.66	10,800.04	47.94%	2,126.34	10,800.03	63.66%
A-n54-k7	1,976.92	10,800.02	22.36%	**1,917.46**	10,800.02	**19.86%**	2,348.71	10,800.05	45.64%	2,199.64	10,800.01	53.53%
A-n55-k9	1,724.69	10,800.02	16.20%	**1,691.86**	10,800.04	**14.49%**	2,332.74	10,800.07	45.81%	1,936.89	10,800.02	63.81%
A-n60-k9	**2,139.07**	10,800.05	**19.80%**	2,163.84	10,800.01	20.29%	2,980.34	10,800.03	57.03%	2,518.18	10,800.03	71.60%
A-n61-k9	-	10,800.01	NOT	**2,261.62**	10,800.05	**38.07%**	2,468.56	10,800.04	50.16%	2,280.26	10,800.02	70.93%
A-n62-k8	**2,073.24**	10,800.04	19.25%	2,074.64	10,800.02	**19.23%**	2,557.78	10,800.04	48.07%	2,598.19	10,800.01	70.34%
A-n63-k10	2,085.93	10,800.03	18.87%	**2,068.76**	10,800.55	**17.59%**	3,111.13	10,800.11	56.52%	2,426.27	10,800.02	72.36%
A-n63-k9	2,836.60	10,800.04	**29.17%**	3,205.54	10,800.68	37.36%	3,321.48	10,800.08	54.52%	**2,830.97**	10,800.06	63.05%
A-n64-k9	**2,128.20**	10,800.01	**17.06%**	2,217.36	10,800.04	20.45%	2,920.31	10,800.05	51.16%	2,545.76	10,800.01	73.14%
A-n65-k9	2,268.71	10,800.02	29.93%	**2,009.80**	10,800.03	**20.96%**	3,019.32	10,800.95	54.50%	2,720.67	10,800.01	76.95%
A-n69-k9	2,108.07	10,800.03	20.76%	**2,087.98**	10,800.08	**20.24%**	3,271.17	10,800.05	51.62%	2,642.48	10,800.03	76.46%
A-n80-k10	2,905.71	10,800.01	20.72%	**2,879.86**	10,800.51	**19.90%**	4,298.03	10,800.08	54.74%	3,350.50	10,800.05	74.82%
mean	**1,683.01**	10,800.02	**15.80%**	1,705.92	10,800.09	16.47%	2144.51	10,800.07	40.22%	1936.53	10,800.02	52.86%
Best	11	-	9	**17**	-	**18**	0	-	0	1	-	0

We further compare Model 2 and the MIPs available in literature (W. Li et al., 2020; L. Liu, Kunpeng Li, and Z. Liu, 2017) according to the relative distance to $CVRP$ optimal solution, which is shown in Table 4.2. The first column presents the instance name, then for each formulation the relative distance to lower bound computed in column LB_CVRP. Column LB_CVRP indicates the objective function of a feasible solution for VRP-Rd created from the optimal CVRP solution, when reported in the literature, by adding the corresponding release date to each customer and letting each vehicle to depart from the depot after the largest release date among all the customer visited on each route. The distance is computed as $(Z_{ip} - LB\_CVRP)/Z_{ip}\%$, thus, this value denotes the Gap from the optimal CVRP solution. A sign "-" in columns LB_CVRP and LR represents that the model was either unable to obtain a feasible solution or the given value is not reported.

As shown in Table 4.2, Model 2 outperforms the two other models in all instances where a CVRP solution configuration is available (19 out of 19). This emphasizes that our Model 2 finds better quality solutions than the other formulations.

Table 4.2: Relative distance to $CVRP$ on VRP-Rd for our both models vs formulation from L. Liu, Kunpeng Li, and Z. Liu, 2017 and W. Li et al., 2020

Instance	Model 2	Three-index formulation (L. Liu, Kunpeng Li, and Z. Liu, 2017)	Two-index formulation (W. Li et al., 2020)	LB_CVRP
A-n32-k5	**-0.85%**	7.60%	3.14%	1225.00
A-n33-k5	**-2.36%**	1.73%	4.12%	1064.32
A-n33-k6	**-1.09%**	-0.94%	7.56%	1060.35
A-n34-k5	**-0.33%**	7.89%	3.93%	1206.11
A-n36-k5	**-0.05%**	12.49%	7.03%	1207.76
A-n37-k5	**-1.28%**	12.67%	9.68%	1028.96
A-n37-k6	**-1.00%**	12.43%	5.78%	1354.93
A-n38-k5	**-0.71%**	9.29%	6.94%	1228.49
A-n39-k5	**-1.21%**	11.21%	6.47%	1316.19
A-n39-k6	**-0.82%**	12.50%	11.90%	1349.91

Table 4.2: Relative distance to $CVRP$ on VRP-Rd for our both models vs formulation from L. Liu, Kunpeng Li, and Z. Liu, 2017 and W. Li et al., 2020 (Cont.)

Instance	Model 2	Three-index formulation (L. Liu, Kunpeng Li, and Z. Liu, 2017)	Two-index formulation (W. Li et al., 2020)	LB_CVRP
A-n44-k7	-	-	-	-
A-n45-k6	**-1.10%**	24.68%	19.23%	1565.27
A-n45-k7	**1.57%**	16.65%	10.24%	1668.20
A-n46-k7	**-1.71%**	19.25%	16.38%	1479.14
A-n48-k7	**0.49%**	15.72%	13.89%	1664.58
A-n53-k7	**-0.48%**	30.09%	20.80%	1683.99
A-n54-k7	**0.81%**	19.02%	13.53%	1901.92
A-n55-k9	**-1.54%**	26.35%	11.30%	1717.97
A-n60-k9	-	-	-	-
A-n61-k9	**24.42%**	30.76%	25.04%	1709.34
A-n62-k8	-	-	-	-
A-n63-k10	-	-	-	-
A-n63-k9	-	-	-	-
A-n64-k9	-	-	-	-
A-n65-k9	**3.37%**	35.68%	28.62%	1942.09
A-n69-k9	-	-	-	-
A-n80-k10	-	-	-	-
mean	**0.85%**	16.06%	11.87%	1440.76
Best	**19**	0	0	-

4.3 Quality of the solutions

In the previous section, we show that the proposed formulation surpass the model reported in the literature, now we shall investigate its performance in terms of the quality of solutions generated against a state-of-the-art algorithm reported in the literature. We compare the results obtained by Model 2, and the ones reported in L. Liu, Kunpeng Li, and Z. Liu, 2017 using GTS (Granular Tabu Search). Since neither the proposed algorithm nor the instances used are publicity available, we make a comparison using the mean. ibid. uses instances generated from well-known CVRP benchmark instances. Still, the release times are generated randomly, as described in Section 4.1. Thus, the generation of the release dates follows a stochastic generation process; thereby, the results presented in ibid. are just a random sample of the process mentioned above. If we draw a large number of samples (generation of release dates), say 30, of the same benchmark, we can obtain appropriate metrics such as the mean, but we could also compute confidence intervals. Having that in mind, we decided to contrast our mean results with the sample reported in ibid.

Table 4.3 compares the results obtained by Model 2 with those reported in ibid. on the same instances. The first column of this table indicates the instance name, columns 2 - 7 deliver the average results, over 30 replicas, for each benchmark instance using Model 2, while columns 8 - 9 show the reported solutions for the instance solved using GTS as presented in ibid. For our model, columns Z_{ip}, T and Gap display the mean of i) the best integer solutions found, ii) the time in seconds required to solve each replica to optimality, and iii) the Gap, respectively. Additionally, we add columns: Min which shows the minimum integer solution found; also we add the upper and lower bounds of the 95% confidence interval (CI), on columns Low and $Upper$, respectively.

For ibid.'s algorithm, we present the best objective function reported and time in seconds to reach that value on columns Z_{ip} and T. Moreover, the penultimate row shows the average of each column and the last row *Best* tallies the number of times each model outperformed the other when comparing for columns Z_{ip} and Gap. A sign "-" in the last row indicates that the given value is not reported. Bold characters in the Z_{ip}, Gap columns mean that, when compared with to other the models, the corresponding instance/mean solution is better. Only `A-n61-k9*` could not yield a feasible solution in every replica; the model obtains solutions in 29 out of 30 replicas.

According to the above table, it is noteworthy that in most instances, our model yield better means solutions than the GTS algorithm presented in L. Liu, Kunpeng Li, and Z. Liu, 2017. We outperform the GTS algorithm in 15 out of 27 instances. Furthermore, it is noticeable that in 25 out of 27, the minimum objective value found among all replicas outperforms the other approach. Additionally, we can observe that in 7 out of the 27 replicas the solution obtain by GTS fall inside our confidence interval; more specifically, eleven times GTS is above the interval and in nine is below. This information is also depicted in Figure 4.1, where the x-axis contains the name of the instance, while the y-axis shows the value of the objective function. Here, except for two cases, the value reported in GTS is within the CI of the 30 runs. This finding reaffirms that our straightforward implementation is equivalent to the cumbersome GTS algorithm. However, our average results are more robust than the sample reported by ibid.

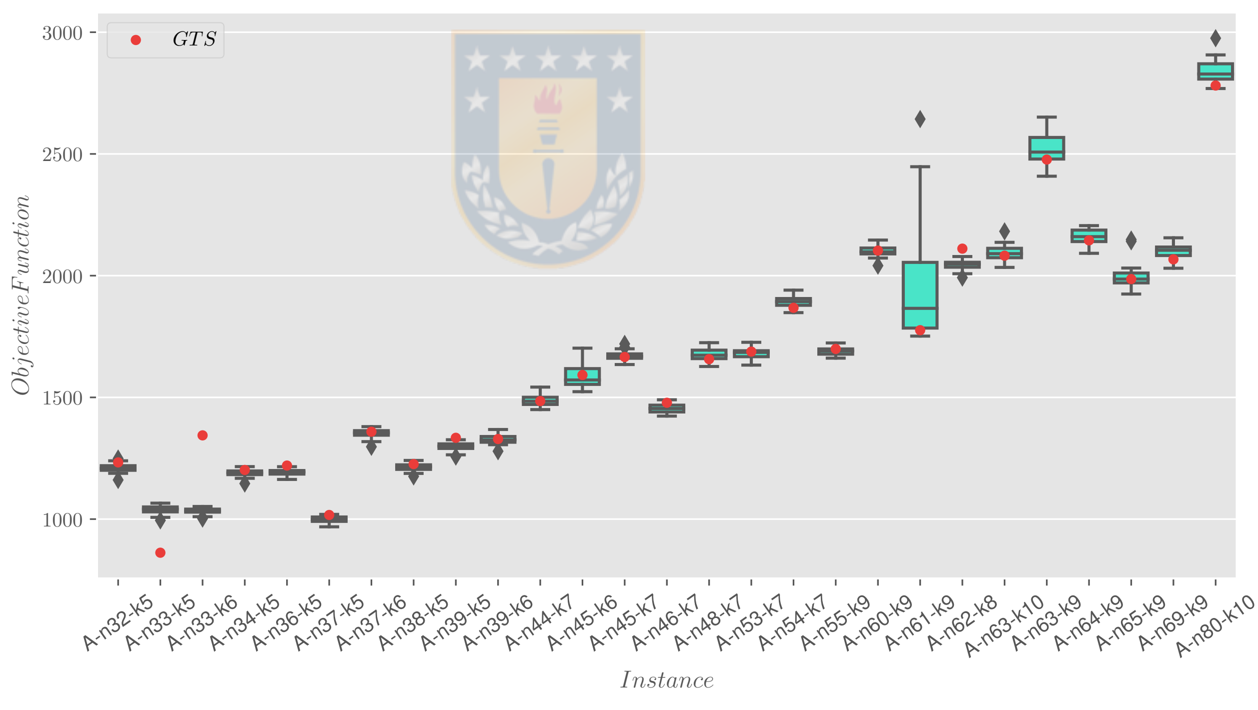


Figure 4.1: Comparison Boxplot between 30 replicas and algorithms presented in L. Liu, Kunpeng Li, and Z. Liu, 2017

Table 4.3: Results of Model 2 with 30 replicas against GTS algorithm presented in L. Liu, Kunpeng Li, and Z. Liu, 2017

Instance	Model 2						GTS	
	Z_ip	T	Gap	Min	Low	$Upper$	Z_{ip}	T
A-n32-k5	**1209.71**	10800.02	10.09%	**1160.86**	**1203.42**	**1216.00**	1233.00	3.00
A-n33-k5	1038.16	10800.02	11.94%	994.82	1032.58	1043.74	**862.00**	3.00
A-n33-k6	**1032.17**	10800.02	9.08%	**1002.32**	**1027.79**	**1036.55**	1344.00	3.00
A-n34-k5	**1190.04**	10800.02	12.35%	**1146.15**	**1185.06**	**1195.01**	1202.00	3.00
A-n36-k5	**1191.84**	10800.02	11.42%	**1162.98**	**1187.52**	**1196.16**	1220.00	3.00
A-n37-k5	**998.00**	10800.02	9.83%	**968.38**	**993.77**	**1002.24**	1017.00	3.00
A-n37-k6	**1352.85**	10800.02	12.94%	**1297.12**	**1346.69**	1359.01	1359.00	3.00
A-n38-k5	**1212.07**	10800.02	15.23%	1175.45	**1207.00**	**1217.14**	1226.00	3.00
A-n39-k5	**1298.13**	10800.02	15.44%	**1256.89**	**1292.48**	**1303.79**	1334.00	4.00
A-n39-k6	**1327.29**	10800.02	14.42%	**1278.90**	**1320.61**	1333.97	1329.00	4.00
A-n44-k7	1485.62	10800.02	15.87%	**1449.85**	**1477.48**	1493.76	**1485.00**	5.00
A-n45-k6	**1588.55**	10800.02	19.06%	**1523.44**	**1571.15**	1605.95	1592.00	5.00
A-n45-k7	1670.51	10800.02	14.43%	**1634.92**	**1663.98**	1677.04	**1666.00**	5.00
A-n46-k7	**1455.28**	10800.02	13.86%	**1423.17**	**1448.54**	**1462.01**	1478.00	5.00
A-n48-k7	1675.28	10800.02	16.24%	**1626.91**	1666.33	1684.23	**1657.00**	5.00
A-n53-k7	**1679.82**	10800.03	18.11%	**1632.57**	**1671.81**	**1687.84**	1688.00	7.00
A-n54-k7	1893.60	10800.03	18.92%	**1848.17**	1885.75	1901.45	**1868.00**	7.00
A-n55-k9	**1689.64**	10800.03	16.08%	**1661.16**	**1683.96**	**1695.32**	1699.00	7.00
A-n60-k9	**2101.17**	10800.07	17.85%	**2041.20**	**2093.19**	2109.16	2103.00	9.00
A-n61-k9*	1954.36	10800.03	33.04%	**1751.74**	1871.71	2037.02	**1776.00**	10.00
A-n62-k8	**2043.15**	10800.07	18.65%	**1992.07**	**2036.31**	**2049.98**	2111.00	9.00
A-n63-k10	2093.73	10800.12	18.53%	**2033.93**	2083.30	2104.16	**2082.00**	10.00
A-n63-k9	2524.20	10800.16	19.89%	**2408.50**	2499.09	2549.31	**2477.00**	10.00
A-n64-k9	2159.01	10800.17	18.41%	**2091.81**	2147.51	2170.51	**2145.00**	10.00
A-n65-k9	1994.22	10800.03	20.73%	**1924.52**	**1976.43**	2012.00	**1986.00**	10.00

Table 4.3: Results of Model 2 with 30 replicas against GTS algorithm presented in L. Liu, Kunpeng Li, and Z. Liu, 2017 (Cont.)

Instance	Model 2						GTS	
	Z_ip	T	Gap	Min	Low	$Upper$	Z_{ip}	T
A-n69-k9	2102.11	10800.03	21.08%	**2030.50**	2091.22	2113.01	**2067.00**	12.00
A-n80-k10	2840.67	10800.03	19.70%	**2768.69**	2823.94	2857.41	**2781.00**	20.00
mean	1659.30	10800.04	16.42%	1603.22	1647.73	1670.88	1658.78	6.59
Best	**15**	-	-	25	18	11	12	-

4.4 Sensitivity analysis: The α parameter

Understanding that the objective function can be read as the merge of two different objectives, where the first corresponds to the minimization of the construction of an optimal set of journeys for the available vehicles, while the second to minimizes the release time of each vehicle. Thus, in the original formulation both were considered as equally important, but we can weight them as follows:

$$\textbf{minimize } \alpha \sum_{(i,j)\in A, i\neq j} d_{ij}x_{ij} + (1-\alpha)\sum_{d\in D} u_d \tag{4.1}$$

Where now α can be set between 0 and 1.

With the addition of a weighted bi-objective function, via the use of an α parameter, we can now test how sensitive our base and lifted models are to the parameter, and how they compare to ibid., when α takes values 0.7 and 0.9. Unfortunately, the inclusion of this objective function on the model presented in W. Li et al., 2020 would change its nature, thereby it will not be compared in this section.

Results are displayed in the Tables 4.4 and 4.5, where the first column indicates the instance name, columns 2 - 4 show the results for each instance tested in our Model 1, columns 5 - 7 inform the outcomes for our Model 2, while columns 8 - 10 communicate the respective solutions for the instances according to the L. Liu, Kunpeng Li, and Z. Liu, 2017's formulation. Columns Z_{ip}, T and Gap convey, respectively, the best integer solution found, the time in seconds to solve the problem to optimality and the relative optimality gap percentage inform by `Gurobi`. Moreover, the penultimate presents the mean for each column, whereas the last row *Best* tallies the number of times each model outperformed the others according to the gap value. A sign "-" in last row indicates that the given value is not reported. Bold characters in the Z_{ip} and Gap columns, and the last row mean when that compared to the other models, the corresponding instance/mean solution is better. Now, on columns T, it means that an optimal solution has been found. Now, from Table 4.4 we can see that when the α parameter is set equal to 0.7, in 14 out of 27 instances our lifted model is better than the other two formulations, while our Model 1 outperforms the others in 13 opportunities. In addition, we can observe that getting α to 0.7 allows the basic model to find a solution for the `A-n61-k9` instance.

If we compare this results with the ones in Table 4.1 (where $\alpha = 0.5$), we observe that the number of best gaps found by the basic model has increased. Previously, our Model 2 outperformed 18 to 9 the vanilla model, while in this weighted case our lifted constraint outperforms our Model 1 only 14 to 13. Also, in Table 4.1, due to the unsolved instance, the basic model obtained an average gap of 15.80% while the lifted model converged to a gap of 16.47%. In the new scenario (where $\alpha = 0.7$) the gaps decreased to 11.14% and 10.39%, respectively. Also, it is relevant to notice that for the models with fewer nodes (up to 40), our vanilla model has a better performance; however, our Model 2 keeps obtaining a better mean objective function.

Now, when we set α to 0.9 (see Table 4.5), which means that generation of routes is more important than minimizing the departure times, we observe an overwhelming change in favor of our vanilla model, since it manages to obtain better gaps in 22 of 27 instances, even delivering 5 optimal solutions. On the other hand, even though our model with lifted constraints manages to reach the same 5 optimal values, it only certify optimality in the first instance. As for the average gaps, we can observe a technical tie, with 6.24% and 6.26% for our basic and our lifted models, respectively. Nevertheless, the Model 2 outperforms the vanilla model by yielding an overall better mean objective function.

Table 4.4: Sentivity analysis: results on VRP-Rd for both models, when $\alpha = 0.7$.

Instance	Model 1			Model 2			L. Liu, Kunpeng Li, and Z. Liu, 2017		
	Z_{ip}	T	Gap	Z_{ip}	T	Gap	Z_{ip}	T	Gap
A-n32-k5	673.40	10800.02	**4.28%**	673.40	10800.02	5.08%	730.99	10800.02	26.87%
A-n33-k5	571.92	10800.01	**4.62%**	573.32	10800.02	5.00%	606.09	10800.02	23.30%
A-n33-k6	603.70	10800.02	**4.04%**	603.70	10800.01	4.99%	610.00	10800.02	20.20%
A-n34-k5	666.03	10800.01	**5.33%**	670.01	10800.02	6.95%	722.73	10800.01	26.97%
A-n36-k5	680.04	10800.01	7.12%	674.93	10800.02	**6.91%**	825.11	10800.01	40.10%
A-n37-k5	566.67	10800.01	**4.98%**	567.19	10800.00	5.78%	613.83	10800.01	21.88%
A-n37-k6	778.16	10800.01	**7.19%**	782.71	10800.00	7.57%	910.38	10800.01	42.01%
A-n38-k5	653.55	10800.01	**5.72%**	653.55	10800.02	6.24%	737.71	10800.03	32.14%
A-n39-k5	719.30	10800.01	8.49%	720.31	10800.01	**8.45%**	867.86	10800.03	38.19%
A-n39-k6	729.67	10800.02	**7.28%**	729.67	10800.01	8.32%	801.11	10800.03	30.70%
A-n44-k7	817.59	10800.02	9.48%	816.03	10800.02	**9.23%**	973.96	10800.02	33.56%
A-n45-k6	990.76	10800.01	23.74%	923.67	10800.03	**18.18%**	1163.58	10800.01	44.90%
A-n45-k7	978.54	10800.02	10.77%	952.63	10800.03	**7.57%**	1064.19	10800.03	38.35%
A-n46-k7	800.08	10800.01	**6.95%**	802.20	10800.01	7.12%	926.55	10800.01	34.06%
A-n48-k7	935.28	10800.02	12.00%	937.34	10800.02	**11.64%**	1156.01	10800.04	44.90%
A-n53-k7	935.14	10800.01	13.66%	903.28	10800.04	**11.28%**	1211.64	10800.05	45.15%
A-n54-k7	1028.79	10800.03	**12.33%**	1033.18	10800.03	12.74%	1252.10	10800.05	47.90%
A-n55-k9	956.33	10800.01	11.72%	933.31	10800.02	**9.77%**	1166.84	10800.03	42.62%
A-n60-k9	1166.06	10800.08	11.95%	1159.00	10800.02	**11.55%**	1557.60	10800.03	57.85%
A-n61-k9	999.28	10800.03	**19.36%**	1093.55	10800.01	26.18%	1442.99	10800.08	55.75%
A-n62-k8	1140.72	10800.11	13.39%	1137.65	10800.02	**12.70%**	1554.60	10800.06	55.51%
A-n63-k10	1147.79	10800.07	**12.77%**	1162.98	10800.02	14.33%	1599.60	10800.06	54.92%
A-n63-k9	1441.15	10800.09	15.75%	1418.96	10800.01	**14.03%**	1943.87	10800.12	58.17%
A-n64-k9	1223.86	10800.22	13.73%	1205.10	10800.02	**12.71%**	1808.44	10800.05	56.90%
A-n65-k9	1181.59	10800.04	21.81%	1046.13	10800.04	**11.95%**	1779.32	10800.06	58.95%
A-n69-k9	1088.21	10800.02	**13.36%**	1098.15	10800.02	14.33%	1601.09	10800.07	49.36%
A-n80-k10	1671.44	10800.03	18.86%	1569.62	10800.02	**13.47%**	2388.80	10800.01	57.06%
mean	931.30	10800.03	11.14%	900.87	10800.02	10.39%	1185.81	10800.04	42.16%
Best	-	-	13	-	-	**14**	-	-	0

Table 4.5: Sentivity analysis: results on VRP-Rd for both models, when $\alpha = 0.9$.

	Model 1			Model 2			L. Liu, Kunpeng Li, and Z. Liu, 2017		
Instance	Z_{ip}	T	Gap	Z_{ip}	T	Gap	Z_{ip}	T	Gap
A-n32-k5	**739.80**	**926.20**	**0.00%**	**739.80**	6083.68	**0.00%**	831.13	10800.01	29.83%
A-n33-k5	**623.97**	**3839.34**	**0.00%**	**623.97**	10800.04	1.96%	643.77	10800.01	21.59%
A-n33-k6	**685.23**	**5221.51**	**0.01%**	**685.23**	10800.01	1.92%	703.49	10800.02	22.96%
A-n34-k5	733.11	10800.01	**1.80%**	736.67	10800.01	3.77%	773.81	10800.04	26.53%
A-n36-k5	750.98	10800.00	**3.03%**	750.98	10800.01	3.65%	827.98	10800.02	34.96%
A-n37-k5	**626.40**	**8892.39**	**0.00%**	**626.40**	10800.01	1.90%	643.1	10800.03	17.13%
A-n37-k6	884.79	10800.02	**4.37%**	884.79	10800.02	4.91%	1018.05	10800.02	39.94%
A-n38-k5	**696.95**	**8504.12**	**0.00%**	**696.95**	10800.05	1.95%	789.96	10800.02	35.98%
A-n39-k5	781.62	10800.00	**4.19%**	782.52	10800.01	5.41%	858.4	10800.01	33.99%
A-n39-k6	787.19	10800.02	**3.68%**	789.89	10800.00	5.38%	803.58	10800.01	26.76%
A-n44-k7	884.71	10800.01	**4.58%**	893.71	10800.02	5.67%	1025.83	10800.01	31.50%
A-n45-k6	937.82	10800.02	**12.61%**	985.61	10800.03	13.69%	1089.35	10800.04	38.43%
A-n45-k7	1072.52	10800.01	**5.12%**	1076.22	10800.02	5.57%	1307.62	10800.03	45.38%
A-n46-k7	863.17	10800.01	**2.64%**	864.07	10800.03	3.35%	1004.42	10800.03	36.24%
A-n48-k7	1038.72	10800.01	**8.17%**	1039.93	10800.02	**8.17%**	1323.46	10800.03	47.58%
A-n53-k7	969.30	10800.01	**5.04%**	967.77	10800.03	6.68%	1346.1	10800.02	47.80%
A-n54-k7	1111.43	10800.01	7.66%	1108.10	10800.06	**7.46%**	1630.11	10800.03	58.71%
A-n55-k9	1007.87	10800.01	**4.77%**	1012.00	10800.02	6.04%	1293.12	10800.04	45.96%
A-n60-k9	1293.30	10800.00	**8.08%**	1296.48	10800.03	9.53%	1732.58	10800.06	61.40%
A-n61-k9	1149.77	10800.01	22.87%	1008.55	10800.03	**9.51%**	1588.91	10800.07	58.50%
A-n62-k8	1250.98	10800.01	**9.04%**	1261.49	10800.01	10.10%	1646.76	10800.03	56.17%
A-n63-k10	1256.04	10800.07	8.28%	1241.98	10800.02	**7.71%**	1881.35	10800.1	59.56%
A-n63-k9	1657.69	10800.01	14.39%	1539.09	10800.03	**7.58%**	2363.46	10800.03	63.05%
A-n64-k9	1330.53	10800.01	**7.63%**	1336.20	10800.04	8.52%	1900.63	10800.01	55.31%
A-n65-k9	1128.41	10800.01	**11.38%**	1185.40	10800.03	11.73%	1872.76	10800.03	58.68%
A-n69-k9	1143.17	10800.01	**8.99%**	1141.75	10800.05	9.00%	1609.05	10800.02	48.99%
A-n80-k10	1750.98	10800.00	10.28%	1722.46	10800.02	**9.31%**	2589.2	10800.03	58.50%
mean	1005.79	9814.22	6.24%	979.19	10618.63	6.26%	1299.92	10800.03	43.02%
Best	**5**	**5**	**22**	**5**	0	6	0	0	0

4.5 Improvement Schemes

In our search for better results, but at the same time with the philosophy of an easy implementation, our next step was to achieve possible improvement schemes for our model. We now take into consideration our lifted model with an alpha of 0.5, which for this given alpha showed better performance in the previous analyses, and we try to improve it. A first approach is the insertion of a feasible initial solution for this we used the implementation of the classic savings algorithm (Clarke and Wright, 1964) included in the free use package ortools from google (Perron and Furnon, 2019). Then, a repair method is used to add the respective variables to convert the CVRP solution into one for VRP-Rd. Additionally, we try with guided local search method to improve this initial solution, with a limit of 600 seconds or until we find 100 possible solutions. The results are shown in the Table 4.6. In this Table, the first column indicates the instance name, columns 2 - 4 (Lifted) deliver the results for each instance for our Model 1, columns 5 - 7 (Initial Solution) inform the outcomes for our Model 1 with the insertion of an initial solution, columns 8 - 10 (Initial Solution Plus Improvement) convey the results for our Model 1 with the insertion of an initial solution plus the use of an guided local search, while columns 11 - 13 (LISPI: Lifted with Initial Solution Plus Improvement) communicate the respective solutions for our Model 2 with the insertion of an initial solution plus the use of an guided local search. Columns Z_{ip}, T and Gap convey, respectively, the best integer solution found, the time in seconds to solve the problem to optimality or relative optimality gap percentage and the gap inform by `Gurobi`. Moreover, on the penultimate row is the mean for each value reported and in the last row the *Best* values found comparing the corresponding models for columns Gap and Z_{ip}. A sign "-" in last row represents that the given value is not reported. Bold characters in Z_{ip} and Gap columns; and the last row means that compare with the other models, for a given instance/mean that solution is better, while on columns T means that an optimal solution has been found and in case more than one model yield the optimal solution T will indicate which one is faster. On Instance column the instances that are follow by a "*" sign as `A-n33-k6*` means that for the respective instance or-tools could not yield a solution and therefore, the instance undergoes a normal optimization of either the our vanilla or our lifted model. From the results presented in the table we can see how the one that has the best performance relative to the average gap and average target value is the Initial Solution Plus Improvement model, with 15.14% and 829.25, respectively, even improving the results of our Lifted with Initial Solution

Plus Improvement model that we would have expected to be better. The first exceeded the others by 9 out of 27 intents, in terms of gap, followed very closely by 8 out of 27 for our lifted model. As for who gets better objective functions, the model with initial solution manages to get the best objective value in 12 out of 27 opportunities. We can conclude that the data help us to see a not so obvious dominance of a model over the others, in fact, for instances with less than 50 nodes it is difficult to say which model has better gaps, even when the Initial Solution model to obtain more times the best objective value; on the other hand, for bigger instances, the indicated model seems to be Initial Solution Plus Improvement.

Table 4.6: Comparison of 4 Improvement schemes to the compact formulation.

Instance	Model 2			Initial Solution			Initial Solution Plus Improvement			LISPI		
	Z_{ip}	T	Gap	Z_{ip}	T	Gap	Z_{ip}	T	Gap	Z_{ip}	T	Gap
A-n32-k5	607.36	10800.01	7.65%	**607.00**	10800.03	8.08%	607.36	10800.02	7.65%	607.36	10800.02	**7.64%**
A-n33-k5	**519.86**	10800.01	8.30%	519.87	10800.01	9.84%	520.36	10800.00	8.65%	**519.86**	10800.03	**8.23%**
A-n33-k6*	**524.47**	10800.01	8.54%	**524.47**	10800.02	8.80%	**524.47**	10800.02	**8.53%**	**524.47**	10800.01	8.54%
A-n34-k5	601.05	10800.01	10.14%	**597.56**	10800.01	**9.86%**	601.05	10800.01	10.76%	601.55	10800.01	10.41%
A-n36-k5	603.57	10800.01	11.21%	599.99	10800.01	**9.55%**	**599.98**	10800.02	9.53%	**599.98**	10800.01	10.06%
A-n37-k5*	507.98	10800.02	9.54%	**499.42**	10800.03	**7.31%**	507.98	10800.06	9.52%	507.98	10800.02	9.52%
A-n37-k6	670.77	10800.02	**11.09%**	674.05	10800.01	12.21%	670.77	10800.02	11.63%	**672.46**	10800.01	11.79%
A-n38-k5	609.94	10800.02	11.68%	**607.70**	10800.01	12.40%	612.69	10800.00	12.18%	609.55	10800.01	**10.36%**
A-n39-k5	**650.24**	10800.01	**14.07%**	655.12	10800.02	14.25%	678.18	10800.02	17.50%	677.35	10800.01	15.44%
A-n39-k6*	669.45	10800.01	**10.53%**	**668.30**	10800.02	11.24%	669.45	10800.03	10.55%	669.45	10800.04	10.54%
A-n44-k7	752.83	10800.01	15.79%	**742.59**	10800.01	**15.08%**	756.35	10800.01	16.82%	751.83	10800.01	16.09%
A-n45-k6	**774.14**	10800.02	18.15%	825.96	10800.02	22.16%	787.14	10800.01	18.23%	783.59	10800.01	**17.67%**
A-n45-k7	847.43	10800.02	14.78%	**840.24**	10800.01	**13.99%**	849.26	10800.01	14.81%	841.72	10800.02	14.38%
A-n46-k7*	727.15	10800.01	**10.12%**	**726.82**	10800.03	10.25%	727.15	10800.03	**10.12%**	727.15	10800.02	**10.12%**
A-n48-k7	836.40	10800.02	17.44%	861.73	10800.01	19.69%	**823.86**	10800.01	**15.89%**	858.51	10800.02	19.37%
A-n53-k7	**837.96**	10800.01	**17.20%**	868.58	10800.02	20.36%	846.32	10800.01	18.03%	841.22	10800.02	17.67%
A-n54-k7	958.73	10800.01	**19.86%**	**957.23**	10800.01	19.89%	961.53	10800.02	20.40%	958.03	10800.03	19.91%
A-n55-k9	**845.93**	10800.01	**14.48%**	853.17	10800.01	15.05%	858.64	10800.02	15.67%	857.18	10800.01	15.36%
A-n60-k9	1081.92	10800.03	20.29%	1132.61	10800.01	23.79%	1069.45	10800.02	19.17%	**1048.03**	10800.03	**17.31%**
A-n61-k9	1130.81	10800.03	38.09%	905.28	10800.02	22.67%	**896.41**	10800.01	**21.91%**	902.48	10800.03	22.37%
A-n62-k8	1037.32	10800.02	19.23%	1047.10	10800.01	20.13%	**1017.81**	10800.01	**17.67%**	1041.31	10800.32	19.57%
A-n63-k10	**1034.38**	10800.74	**17.59%**	1073.67	10800.04	21.38%	1047.16	10800.13	18.98%	1074.66	10800.02	21.34%
A-n63-k9	1602.77	10800.83	37.34%	1268.56	10800.11	20.81%	**1235.31**	10800.04	**18.59%**	1240.75	10800.01	18.91%
A-n64-k9*	1108.68	10800.05	20.45%	1089.82	10800.02	19.26%	**1079.73**	10800.02	**18.29%**	1133.63	10800.03	22.25%
A-n65-k9	1005.90	10800.00	21.10%	995.12	10800.04	20.33%	**975.52**	10800.06	**18.51%**	1030.90	10800.02	23.06%
A-n69-k9	1045.99	10800.02	20.40%	**1039.25**	10800.00	**19.80%**	1046.56	10800.06	20.41%	1071.88	10800.02	22.27%
A-n80-k10	1440.93	10800.07	20.02%	**1414.48**	10800.04	18.79%	1419.15	10800.05	**18.70%**	1441.38	10800.18	20.16%
mean	853.11	10800.07	16.48%	836.88	10800.02	15.81%	**829.25**	10800.03	**15.14%**	836.82	10800.04	15.57%
Best	7	-	8	**12**	-	6	9	-	**9**	5	-	6

4.6 Commercial Solver parameter tunning

For an analysis of `Gurobi`'s internal parameters, we took 100 instances of those generated in the replicas, that is, of the 27 original problems with different seeds for the generation of release dates, and we tested with the `tune()` method, giving `Gurobi` a time limit of the process of three days of resolution, a time limit per instance of 30 minutes (1800 seconds) and 15 runs per set of parameters, to eliminate possible errors due to seed selection. From these results it was obtained that in 89 out of 100 instances the base parameters were improved. From the base parameters we obtain an average gap (here we will only consider the instances in which it was possible to improve the base parameters) of 19.83% against 18. 57% when compared with the improved parameters, now if we consider that there are disturbing effects by adding to these calculations the instances in which it was not possible to solve the 15 runs per sample, which in this case are 9 instances out of the 89 in which it is achieved to improve the results, we are left with 80 instances and even so it is observed a decrease from 17.61% to 16.78%, which means a reduction of 4.71% of the mean gaps. Within the parameters that are most repeated in this tuning process are `MIPFocus`, `Heuristics`, `GomoryPasses`, `Aggregate` and `PreDual`, with an appearance of 24, 14, 12, 12 and 10 times, respectively, in the set of improved parameters. Now from these five, if we consider the percentage improvement in mean gaps, we can find improvements of 11.36%, 3.03%, 2.96%, 2.85% and 2.70%, for `Heuristics`, `MIPFocus`, `PreDual`, `GomoryPasses` and `Aggregate`, respectively (Samples with runs that do not obtain full solutions are not considered again in these mean gap improvements.)

As for these samples with runs without integer solution, we can observe a substantial improvement, due to the fact that in the nine instances without solution a solution is not found in 30 out of 135 seeds, on the other hand when comparing it with the set of tuned parameters, this amount drops to 10 seeds. At the same time, the average gap in these instances drops from 39.56% to 34.41%, which reports a percentage decrease of 13.01% (Note that there are instances in which solutions are found in a little more than half of the seeds, while the improved parameters manage to obtain a solution in almost all of them, but reporting higher average gaps).

Finally, in the following box we show the parameters and their (average) values for the previous analysis[2]:

```
Heuristics: 0.5
MIPFocus: 2
PreDual: 1
GomoryPasses: 1
Aggregate: 0
```

This is by no means an exhaustive analysis, it is recommended to tune the parameters using gurobi's own tools.

[2] For a further explanation on these parameter see A.3

Chapter 5

Concluding remarks

In this research, a compact formulation for the vehicle routing problem with release times (VRP-Rd) has been proposed, a variant of the vehicle routing problem in which each customer's order has a release date indicating the earliest time that the order is available at the depot for delivery. Additionally, it has been proposed both valid inequalities and lifted constraints that accelerate the proposed model. Furthermore, it has undergone researches on ease-to-follow improvement schemes and try by tuning `Gurobi`'s parameters.

The computational results suggest that the proposed formation outperforms the existing ones, reported in the literature, for the VRP-Rd (W. Li et al., 2020; L. Liu, Kunpeng Li, and Z. Liu, 2017) when solved directly in a commercial solver. Moreover, after a comprehensive study of our model under different scenarios, it has been empirically demonstrated that it is competitive (in terms of solution quality) with a state-of-the-art algorithm reported in the literature. Another advantage of our model over existing algorithms reported in the literature is that it can be easily implemented by practitioners. Then, it is shown that the formulation can be solved efficiently by a commercial solver without complicated algorithmic implementations. Finally, computational experiments suggest that the insertion of an initial solution can indeed improve the solutions and a further improvement of this initial solution can be beneficial for larger instances.

For future research, it will be intended to extend and adapt our formulation to other routing problems with release dates arising in the literature and in real-life applications. Another potential research direction is to develop customized, exact algorithms via column generation or Branch & Cut to exploit the structure of the proposed formulations.

Bibliography

[1] Olga Andrienko. *Ecommerce & Consumer Trends During Coronavirus*. 2020. URL: https://www.semrush.com/blog/ecommerce-covid-19/ (visited on 10/13/2020).

[2] D.L. Applegate et al. *The Traveling Salesman Problem: A Computational Study*. Princeton, New Jersey: Princeton University Press, 2006.

[3] Claudia Archetti, Dominique Feillet, Andrea Mor, and M Grazia Speranza. "An iterated local search for the traveling salesman problem with release dates and completion time minimization". In: *Computers & Operations Research* 98 (2018), pp. 24–37.

[4] Claudia Archetti, Dominique Feillet, Andrea Mor, and Maria Grazia Speranza. "Dynamic traveling salesman problem with stochastic release dates". In: *European Journal of Operational Research* 280.3 (2020), pp. 832–844.

[5] Claudia Archetti, Dominique Feillet, and M Grazia Speranza. "Complexity of routing problems with release dates". In: *European journal of operational research* 247.3 (2015), pp. 797–803.

[6] Claudia Archetti, Ola Jabali, and M Grazia Speranza. "Multi-period vehicle routing problem with due dates". In: *Computers & Operations Research* 61 (2015), pp. 122–134.

[7] Alp M Arslan et al. "Crowdsourced deliveryA dynamic pickup and delivery problem with ad hoc drivers". In: *Transportation Science* 53.1 (2019), pp. 222–235.

[8] Caesar William Augusta. "ONLINE SHOPPING (E-COMMERCE) IS ONE OF THE ALTERNATIVE TO GO SHOPPING DURING IN PANDEMIC". In: (2020).

[9] M Babagolzadeh et al. “Sustainable open vehicle routing with release-time and time-window: a two-echelon distribution system”. In: *IFAC-PapersOnLine* 52.13 (2019), pp. 571–576.

[10] Alexander Barvinok, Edward Kh Gimadi, and Anatoliy I Serdyukov. “The maximum TSP”. In: *The Traveling Salesman problem and its variations*. Springer, 2007, pp. 585–607.

[11] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. “The vehicle routing problem: State of the art classification and review”. In: *Computers & Industrial Engineering* 99 (2016), pp. 300–313. ISSN: 0360-8352. DOI: https://doi.org/10.1016/j.cie.2015.12.007. URL: http://www.sciencedirect.com/science/article/pii/S0360835215004775.

[12] Pedro Campelo et al. “Consistent vehicle routing problem with service level agreements: A case study in the pharmaceutical distribution sector”. In: *European Journal of Operational Research* 273.1 (2019), pp. 131–145.

[13] Jacques Carlier. “Scheduling jobs with release dates and tails on identical machines to minimize the makespan”. In: *European Journal of Operational Research* 29.3 (1987), pp. 298–306.

[14] Diego Cattaruzza, Nabil Absi, and Dominique Feillet. “The multi-trip vehicle routing problem with time windows and release dates”. In: *Transportation Science* 50.2 (2016), pp. 676–693.

[15] Diego Cattaruzza, Nabil Absi, Dominique Feillet, and Jesús González-Feliu. “Vehicle routing problems for city logistics”. In: *EURO Journal on Transportation and Logistics* 6.1 (2017), pp. 51–79.

[16] Geoff Clarke and John W Wright. “Scheduling of vehicles from a central depot to a number of delivery points”. In: *Operations research* 12.4 (1964), pp. 568–581.

[17] Teodor Gabriel Crainic, Nicoletta Ricciardi, and Giovanni Storchi. “Models for evaluating and planning city logistics systems”. In: *Transportation science* 43.4 (2009), pp. 432–454.

[18] George B Dantzig and John H Ramser. “The truck dispatching problem”. In: *Management science* 6.1 (1959), pp. 80–91.

[19] Maryam Darvish, Leandro C Coelho, and Gilbert Laporte. “The Dynamic Routing Problem with Due Dates and Stochastic Release Dates”. In: (2020).

[20] Martin Desrochers and Gilbert Laporte. “Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints”. In: *Operations Research Letters* 10.1 (1991), pp. 27–36.

[21] Abe Dunn, Kyle Hood, and Alexander Driessen. “Measuring the effects of the COVID-19 pandemic on consumer spending using card transaction data”. In: *US Bureau of Economic Analysis Working Paper WP2020-5* (2020).

[22] Bernhard Fleischmann. *The vehicle routing problem with multiple use of the vehicles*. 1990.

[23] Bezalel Gavish and Stephen C Graves. “The travelling salesman problem and related problems”. In: (1978).

[24] Eitan Israeli and R Kevin Wood. “Shortest-path network interdiction”. In: *Networks: An International Journal* 40.2 (2002), pp. 97–111.

[25] Dmitry Ivanov. “Viable supply chain model: integrating agility, resilience and sustainability perspectiveslessons from and thinking beyond the COVID-19 pandemic”. In: *Annals of Operations Research* (2020), p. 1.

[26] TS Jaikishan and R Patil. “A Reactive GRASP Heuristic Algorithm for Vehicle Routing Problem with Release Date and Due Date Incurring Inventory Holding Cost and Tardiness Cost”. In: *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE. 2019, pp. 1393–1397.

[27] Roy Jonker and Ton Volgenant. “Improving the Hungarian assignment algorithm”. In: *Operations Research Letters* 5.4 (1986), pp. 171–175.

[28] Imdat Kara, Gilbert Laporte, and Tolga Bektas. "A note on the lifted Miller–Tucker–Zemlin subtour elimination constraints for the capacitated vehicle routing problem". In: *European Journal of Operational Research* 158.3 (2004), pp. 793–795.

[29] Mathias A Klapp, Alan L Erera, and Alejandro Toriello. "The dynamic dispatch waves problem for same-day delivery". In: *European Journal of Operational Research* 271.2 (2018), pp. 519–534.

[30] Mathias A Klapp, Alan L Erera, and Alejandro Toriello. "The one-dimensional dynamic dispatch waves problem". In: *Transportation Science* 52.2 (2018), pp. 402–415.

[31] Kai Li and Ba-Yi Cheng. "Variable neighborhood search for uniform parallel machine makespan scheduling problem with release dates". In: *2010 International Symposium on Computational Intelligence and Design*. Vol. 2. IEEE. 2010, pp. 43–46.

[32] Wenli Li et al. "Multi-trip vehicle routing problem with order release time". In: *Engineering Optimization* 52.8 (2020), pp. 1279–1294.

[33] Ling Liu, Kunpeng Li, and Zhixue Liu. "A capacitated vehicle routing problem with order available time in e-commerce industry". In: *Engineering Optimization* 49.3 (2017), pp. 449–465.

[34] Stef Moons et al. "Integration of order picking and vehicle routing in a B2C e-commerce context". In: *Flexible Services and Manufacturing Journal* 30.4 (2018), pp. 813–843.

[35] Andrea Mor and Maria Grazia Speranza. "Vehicle routing problems over time: a survey". In: *4OR* (2020), pp. 1–21.

[36] Laurent Perron and Vincent Furnon. *OR-Tools*. Version 7.2. Google, 2019. URL: https://developers.google.com/optimization/.

[37] Harilaos N Psaraftis, Min Wen, and Christos A Kontovas. "Dynamic vehicle routing problems: Three decades and counting". In: *Networks* 67.1 (2016), pp. 3–31.

[38] Damián Reyes, Alan L Erera, and Martin WP Savelsbergh. "Complexity of routing problems with release dates and deadlines". In: *European journal of operational research* 266.1 (2018), pp. 29–34.

[39] Juan Carlos Rivera, H Murat Afsar, and Christian Prins. "A multistart iterated local search for the multitrip cumulative capacitated vehicle routing problem". In: *Computational Optimization and Applications* 61.1 (2015), pp. 159–187.

[40] Daniel Schubert, André Scholz, and Gerhard Wäscher. "Integrated order picking and vehicle routing with due dates". In: *OR Spectrum* 40.4 (2018), pp. 1109–1139.

[41] Benjamin Shelbourne. "The vehicle routing problem with release and due dates: formulations, heuristics and lower bounds". PhD thesis. University of Southampton, 2016.

[42] Benjamin C Shelbourne, Maria Battarra, and Chris N Potts. "The vehicle routing problem with release and due dates". In: *INFORMS Journal on Computing* 29.4 (2017), pp. 705–723.

[43] David Simchi-Levi. "New worst-case results for the bin-packing problem". In: *Naval Research Logistics (NRL)* 41.4 (1994), pp. 579–585.

[44] Marius M Solomon. "Algorithms for the vehicle routing and scheduling problems with time window constraints". In: *Operations research* 35.2 (1987), pp. 254–265.

[45] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. Vol. 18. SIAM, 2014.

[46] Marlin W Ulmer, Barrett W Thomas, Ann Melissa Campbell, et al. "The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times". In: *Transportation Science* (2020).

[47] Marlin W Ulmer, Barrett W Thomas, and Dirk C Mattfeld. "Preemptive depot returns for dynamic same-day delivery". In: *EURO journal on Transportation and Logistics* 8.4 (2019), pp. 327–361.

[48] Stacy A Voccia, Ann Melissa Campbell, and Barrett W Thomas. "The same-day delivery problem for online purchases". In: *Transportation Science* 53.1 (2019), pp. 167–184.

[49] Baris Yildiz and Martin Savelsbergh. "Provably high-quality solutions for the meal delivery routing problem". In: *Transportation Science* 53.5 (2019), pp. 1372–1388.

Appendix A

VRP-Rd

A.1 Value of Ms

In this section, the derivation of the structural parameters for our model are presented.

A.1.1 Journeys duration constraints

$$u_i + t_{ij} - u_j \leq M_1(1 - x_{ij}),\ \ \forall i, j \in A,\ i \neq j \tag{3.8}$$

Derivation: If $x_{ij} = 0$, then it is obtained $u_i + t_{ij} - u_j \leq M_1$. Thus, the minimum value for u_j occurs when $j \in C$, then $u_j \geq \min_{d \in D} u_d + \min_{d \in D, j \in C} t_{dj} = \min_{i \in C} r_i + \min_{d \in D, j \in C} t_{dj}$. Because the VRP-Rd does not consider a maximum time limit nor time windows, u_i does not have an structural limit parameter, rather it will be defined this maximum as $u_i \leq \max u_f$ and further explain in A.2. By definition, $t_{ij} \leq \max_{i,j \in C} t_{ij}$. Thus, $M_1 = \max u_f + \max_{i,j \in C, i \neq j} t_{ij} - \min_{i \in C} r_i - \min_{d \in D, j \in C} t_{dj}$.

A.1.2 Release date constraints

$$u_d \geq h_i - M_2(1 - x_{di}), \forall d \in D, \forall i \in C \tag{3.9}$$

Derivation: If $x_{di} = 0$, then it is obtained $h_i - u_d \leq M_2$. Then, it is easy to see that any vehicle cannot depart before the release time have been completed, and therefore, $u_d \geq \min_{i \in C} r_i$. Also, by definition $h_i \leq \max_{i \in C} r_i$. Thus, $M_2 = \max_{i \in C} r_i - \min_{i \in C} r_i$

$$h_j \geq h_i - M_3(1 - x_{ij}), \ \ \forall i,\ j \in C, \ \ i \neq j \tag{3.10}$$

Derivation: If $x_{ij} = 0$, then it is obtained $h_i - h_j \leq M_3$. Then, it can be observed how h_j is restricted to be at least $\min_{i \in C} r_i$. Also, by definition $h_i \leq \max_{i \in C} r_i$. Thus, $M_3 = \max_{i \in C} r_i - \min_{i \in C} r_i$

And in a similar fashion it can be also derived the value for M_4

$$h_i \geq h_j - M_4(1 - x_{ij}), \ \ \forall i,\ j \in C, \ \ i \neq j \tag{3.11}$$

Derivation: If $x_{di} = 0$, then it is obtained $h_j - h_i \leq M_4$. Thus, $M = \max_{i \in C} r_i - \min_{i \in C} r_i$

A.2 Calculating $\max u_f$

The issue when calculating $\max u_f$ is first to understand what problem does it refers to. As initially, there is no limit for our problem, for example, a time limit of t_H, virtually the arrival times, i.e., $u_i \forall i \in N$, could be any given number to infinity, then, it is needed to find the upper bound given a VRP-Rd instance. This problem can be visualized as finding a set of routes of maximum distance, such that it pass through all nodes, that leave and return to the depot without violating the maximum capacity of the fleet. This problem could be defined as a maximum VRP, that is, finding one or a set of Hamiltonian cycles without surpassing the fleet capacity. As it can be seen and if it is considered as base the graph with weights $-c$, this problem, except for some similar cases to its related VRP, would be NP-hard and therefore complex to solve for our interest that is to find a value for the big-M. If the restrictions of fleet capacity are relaxed it is obtained the maximum TSP problem, which continues being NP-hard (Barvinok, Gimadi, and Serdyukov, 2007). Then again a constraint is relaxed, in this case the sub-tour elimination constraints, it is obtained a maximum general assignment problem, which can be solved with the Hungarian algorithm with a known complexity of $\mathcal{O}(|N|^3)$ (Jonker and Volgenant, 1986). As a quicker example of possible value for $\max u_f$ it can be set it as the maximum arc times the number of nodes, *i.e.*, $|N| \max_{i,j \in C, i \neq j} t_{ij}$

Then the objective value of these problems should look like:

$$Z_{maximum\ VRP} \leq Z_{maximum\ TSP} \leq Z_{max\ GAP} \leq |N| \max_{i,j \in C, i \neq j} t_{ij}$$

A.3 Parameter used in the tunning of `Gurobi`

For a complete overview of `Gurobi` and its parameters it is highly recommended seeing the documentation available on the website `www.gurobi.com`. The five parameter mention on this paper an their usage are the following:

- `Heuristics`: Time spent in feasibility heuristics. Determines the amount of time spent in MIP heuristics. Type: `double`, default value: 0.05, minimum value: 0, maximum value: 1.

- `MIPFocus`: MIP solver focus. Allows you to modify your high-level solution strategy, depending on your goals. If you are more interested in finding feasible solutions quickly, you can select MIPFocus=1. If you believe the solver is having no trouble finding good quality solutions, and wish to focus more attention on proving optimality, select MIPFocus=2. If the best objective bound is moving very slowly (or not at all), you may want to try MIPFocus=3 to focus on the bound. Type: `int`, default value: 0, minimum value: 0, maximum value: 3.

- `PreDual`: Controls presolve model dualization. Controls whether presolve forms the dual of a continuous model. Depending on the structure of the model, solving the dual can reduce overall solution time. Type: `int`, default value: -1, minimum value: -1, maximum value: 2.

- `GomoryPasses`: Gomory cut passes. A non-negative value indicates the maximum number of Gomory cut passes performed. Type: `int`, default value: -1, minimum value: -1, maximum value: MAXINT.

- `Aggregate`: Presolve aggregation. Enables or disables aggregation in presolve. In rare instances, aggregation can lead to an accumulation of numerical errors. Turning it off can sometimes improve solution accuracy. Type: `int`, default value: 1, minimum value: 0, maximum value: 1.