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

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

Both formulations have been studied independently before – the Arc-Item is very similar to a multi-commodity-flow formulation in Letchford & Salazar-González [42] and the Arc-Load formulation has been studied in Fukasawa et al. [25] – and their LP relaxations are incomparable. Nonetheless, we show that a formulation combining the two (called Arc-Item-Load) may lead to a significantly stronger LP relaxation, thereby indicating that the two formulations capture complementary aspects of the problem. In addition, we study how set partitioning based formulations can be combined with these formulations. We present computational experiments on several well-known benchmark instances that highlight the advantages and drawbacks of the LP relaxation of each formulation and point to potential avenues of future research.

Keywords. Cumulative Vehicle Routing Problem, Arc-Item-Load, Arc-Item, Arc-Load, Set Partitioning.

1 Introduction
One of the most studied versions of the class of Vehicle Routing Problems (VRPs) [31, 57] is the classical Capacitated Vehicle Routing Problem (CVRP) [15]. The CVRP can be described as finding a minimum distance set of K routes that start and end at the depot, picking up the customers’ demands while respecting the capacities of the vehicles. The Cumulative Vehicle Routing Problem (CmVRP), also known as Energy Minimization Vehicle Routing Problem [37, 38], differs from the former by considering an energy consumption measure as its cost to be minimized. In particular, the cost of traversing an arc now becomes the product of the arc’s distance and the load carried while traversing it.

The consideration of the load carried on an arc, in addition to its distance, becomes important as increasing concerns about the environment (and thus the desire to minimize fuel consumption, pollution or related quantities) become more prevalent (see, for example, the surveys Demir et al. [16] and Egele & Bektas [21] on Green VRPs). Indeed, Kara et al. [38] mention the use of CmVRP in such a context as a simplified way to minimize the energy consumed. In addition, the energy consumption function of electric unmanned aerial vehicles, also known as drones, [47, 32] and several other types of vehicles are well modeled by the CmVRP [18, 59, 38]. The CmVRP captures critical aspects of multi-package pickup electric drones, as their total weight will increase along the route, affecting the usually restricted autonomy provided by the battery. Thus, the CmVRP is an increasingly important problem to solve in multiple applications.
Several related combinatorial optimization problems have also been studied in the literature. For instance, the CmVRP generalizes the $K$-Traveling Repairman Problem ($KTRP$) [22, 23, 36], where there are no customer demands and the total cost is given by the sum of the waiting times of the customers. When there is only one repairman, the $KTRP$ is also known as Traveling Repairman Problem (TRP) [2, 3], Traveling Deliveryman Problem [44, 45], Cumulative Traveling Salesman Problem [8], and Minimum Latency Problem [9, 3, 10]. The TRP is also related to the Time Dependent Traveling Salesman Problem (TDTSP) [51, 1, 10]. The version of the CmVRP with a single vehicle was studied by Wang et al. [58], who presented a mathematical model based on Fukasawa et al. [25].

The CmVRP is not only a strongly NP-hard problem, but it has also shown to be quite challenging computationally. The work of Kara et al. [37, 38] initially proposed an exact algorithm for the CmVRP. The best exact results for the problem are from Fukasawa et al. [25], who present both a branch-and-cut (BC) and a branch-and-cut-and-price (BCP) algorithm, where the latter yields the state-of-the-art results. As a matter of fact, the best results for the CmVRP also come from a BCP approach [50]. There have also been heuristic approaches for the CmVRP, for instance by Kramer et al. [39] using a matheuristic and by Zachariadis et al. [60] using a local search approach, as well as by Xiao et al. [59]. A variant of the problem with limited duration was addressed by Cinar et al. [12, 13] using constructive algorithms and metaheuristics, whereas the work of Cinar et al. [13] also surveys some methods applied to solve the CmVRP. Heuristic approaches based on linear programming and column generation for the problem were presented by Gaur & Singh [27] and Singh & Gaur [56]. Without constraining the number of vehicles that can be used, the works of Gaur et al. [28] and Mulati & Miyazawa [46] presented an approximation algorithm for the problem with a factor of less than 4, while Gaur et al. [29, 30] proposed a 6-approximation for the problem with stochastic demands.

While the work of Fukasawa et al. [25] uses arc-load based formulations, whose variables represent if a given weight is carried over a specific arc, the central idea of this work is to combine such a formulation with an arc-item based formulation which keeps track of the individual items that are carried over each arc. We say an arc carries an item if vertex $i$ has been visited prior to traversing the arc. It is worth noting that the arc-item formulation is very similar to multi-commodity-flow formulations with one commodity per vertex, that have been studied for the CVRP (see for instance Letchford & Salazar-González [42, 43]). Our main contribution in this paper is to combine an Arc-Item formulation for the CmVRP with arc-load ones and to study their theoretical and experimental properties. In particular, we show that one such formulation – the Arc-Item coupled to the Arc-Load, i.e., the Arc-Item-Load (AIL) – strictly dominates other previous formulations of the problem. Following standard notation, we say that a formulation $F$ dominates a formulation $G$ if, for any solution of $F$’s Linear Programming (LP) relaxation, there exists a solution of $G$’s LP relaxation with the same objective function value. Throughout this paper, we also use “relaxation” or “relaxed” to denote the LP relaxation of a formulation.

Since set partitioning based formulations tend to be the most successful in several VRPs, and due to an equivalence between arc-load formulations for the CmVRP and a natural set partitioning formulation, we also consider how the Arc-Item formulation can be combined with such set partitioning formulations. We show one such formulation that achieves the same bound as our compact formulation. While this may indicate that the compact formulation is more useful since no column generation needs to be performed, the set partitioning based formulation can be easily strengthened by using more advanced techniques such as $ng$-routes [5], leading to better bounds. We also propose one natural way to extend the arc-item formulation with a set partitioning one, but show that it may become too hard to solve, since its pricing subproblem is strongly NP-hard. Thus, we propose a final formulation attempting to address this drawback.

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

The other sections of this paper are organized as follows. Section 2 presents a literature review of problems and formulations related to our topic. Section 3 contains the precise CmVRP definition and other fundamental concepts. Section 4 presents our proposed formulations and some of their properties. Experimental results are presented in Section 5 and concluding remarks are written in Section 6.

## 2 Literature Review

Several formulations have been proposed for the CVRP, for instance: two-index and three-index vehicle-flow, one-commodity-flow and multi-commodity-flow, and set partitioning based formulations [35, 55]. Except for the set partitioning based ones, the others are classified as compact formulations, i.e., they have a polynomial number of variables. The current best exact algorithms for the CVRP are based on set partitioning formulations combined with families of cutting planes inside a BCP framework [24, 4, 5, 14, 52, 48, 50].
The CmVRP bears many similarities with the CVRP, and most formulations for one of them can be adapted to the other. Indeed, several CVRP formulations have been used in the cumulative problem, for instance one-commodity-flow, arc-load, and set partitioning formulations [38, 25]. Similar to what happens in the CVRP, the best exact approaches to the CmVRP rely on a BCP framework over a set partitioning formulation with additional cutting planes [25]. It is not too surprising that formulations for the CVRP have been adapted to the CmVRP, as the feasible set is the same for both problems, with the only difference arising in how the cost of a solution is calculated. Thus, we devote the rest of this section to reviewing other formulations for the CmVRP.

In the series of papers by Letchford & Salazar-González [41, 42, 43], several mixed integer linear programming formulations for the CVRP are presented and/or proposed. We highlight two particular ones which are related to our work, namely MCF1a and MCF2a, based on a multi-commodity-flow idea. Beyond the binary variables \( x_a \) indicating whether arc \( a \) is used, their MCF2a formulation, whose main concept can be traced back to Garvin et al. [26], contains binary variables to indicate whether an arc \( a \) is traversed by a vehicle before \( ([f_a^0] \text{ variables}) \) or after \( ([g_a^0] \text{ variables}) \) visiting customer \( i \). This kind of formulation also has precedents in modeling the Traveling Salesman Problem (TSP) and some variations [40, 33]. The MCF1a formulation is conceptually similar to the MCF2a, with the difference that only the variables \( x_a \) and \( f_a^0 \) are present. Moreover, the MCF2a model was strengthened by coupling it to a knapsack based formulation with exponentially many columns, each one corresponding to a packing pattern of the items. The pricing of the variables can be done in pseudo-polynomial time and the resulting formulation is called MCF2K.

Also based on the MCF2a, the Arc-Packing (AP) formulation couples the existing variables with a new set of arc-packing binary variables \( [\phi_a^{0, 2}] \), which take the value of one if and only if a vehicle traverses arc \( a \) after having visited the customers (packed) in \( S \), and no more, and about to visit the ones (packed) in \( T \), and no more. As \( S, T, N \) are the set of all customers, the number of all possible columns is exponential, however its pricing subproblem can be solved in pseudo-polynomial time. In their work, Letchford & Salazar-González [43] were also able to couple the AP to a set partitioning formulation aware of the number of times an arc appears in a non-elementary route, and by this, creating the APnSP. The dominance order is given by APnSP, AP, MCF2K, MCF2a, and MCF1a; where the first dominates the second and so on. With a time limit of two hours (for the first four models), they carried out experiments including relaxations of these formulations over new CVRP instances with 16 vertices, divided into 12 families of 20 instances each, which helped to establish the dominance relationships. It is worth mentioning that their work deals with the CVRP version that does not fix the number of vehicles in advance.

### 3 Fundamentals

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

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

The CmVRP One-Commodity-Flow formulation (CF) was introduced alongside the problem itself by Kara et al. [37, 38]. It has two sets of decision variables over the arcs \( a \in A \): binary variables \( x_a \) that indicate whether the arcs are traversed by a vehicle, and the continuous \( y_a \geq 0 \) variables representing the amount of flow passing through the arcs. Conceptually, the flow passing through an arc is equivalent to the load carried on it. The LP relaxation of this formulation is normally solved in a short amount of time. However, the lower bound it provides is weak when compared to the relaxation of the Arc-Load formulation used in Fukasawa et al. [25], and, thus, a branch-and-bound (BB) type algorithm based on the CF formulation tends to explore a larger number of nodes. Thus, we refrain from presenting the CF formulation and present the Arc-Load one next.
3.1 The Arc-Load Formulation

The CmVRP Arc-Load formulation (AL) was proposed by Fukasawa et al. [25]. It uses decision variables that determine if an arc is being used while carrying a specific weight \( \ell \) that is in a discrete set given by \( L = \{0, 1, \ldots, Q\} \). The decision variables for the so-called arc-loads are

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

We note that such a formulation has been used in other similar problems. Indeed it can be seen as a generalization of a formulation of Picard & Queyranne [51] for the TDTSP.

The formulation is presented below.

\[
\text{(AL) } \min \sum_{a \in A} \sum_{\ell \in L} d_a (q^0 + \ell) y_a^{\ell} \\
\text{s.t.} \sum_{a \in \delta^{-}(0)} \sum_{\ell \in L} y_a^{\ell} = K, \quad \forall 0 \in N, \quad (1b)
\]

\[
\sum_{a \in \delta^{+}(0)} \sum_{\ell \in L} y_a^{\ell} = K, \quad \forall 0 \in N, \quad (1c)
\]

\[
\sum_{a \in \delta^{+}(0)} \sum_{\ell \in L} y_a^{\ell} = 1, \quad \forall u \in N, \quad (1d)
\]

\[
\sum_{a \in \delta^{-}(w)} \sum_{\ell \in L} y_a^{\ell} = \sum_{a \in \delta^{+}(w)} y_a^{\ell q^w}, \forall w \in N, \ell \in \{0, \ldots, Q - q^w\}, \quad (1e)
\]

\[
y_{u0}^{\ell} = 0, \quad \forall u \in \delta^{-}(0), \ell \in \{0, \ldots, q^u - 1\}, \quad (1f)
\]

\[
y_a^{\ell} = 0, \quad \forall a \in \delta^{+}(0), \ell \in L \setminus 0, \quad (1g)
\]

\[
y_{uv}^{\ell} = 0, \quad \forall uv \in A \setminus \delta(0), \ell \in \{0, \ldots, q^u - 1, Q - q^v + 1, \ldots, Q\}, \quad (1h)
\]

\[
0 \leq y_a^{\ell} \leq 1, \quad \forall a \in A, \ell \in L, \quad (1i)
\]

\[
y_a^{\ell} \in \mathbb{Z}, \quad \forall a \in A, \ell \in L. \quad (1j)
\]

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

Constraints (1b) and (1c) enforce the in-degree and out-degree of the depot to be equal to the number of vehicles. The out-degree of a customer is restricted to one by (1d). The load balance of the customers is managed by (1e). Constraints (1f) ensure that arcs entering the depot must have at least the load weight of the last visited customer. Constraints (1g) enforce that arcs leaving the depot must carry no load. For the arcs that are not incident to the depot, constraints (1h) block some low and high loads.

In Theorem 1, we retrieve a relation between AL and CF formulations. First, we need to define some notation. The feasible region of the LP relaxation of a given formulation is given by \(\mathcal{P}\) subscribed by the name of the formulation, for instance \(\mathcal{P}_{AL}\) denotes the LP relaxation of AL. Similarly, we define \(\mathcal{P}_{CF}\) subscribed by the name of the formulation to be the value of an optimal solution for the respective relaxed model.

**Theorem 1** (Fukasawa et al. [25]) The formulation AL dominates CF, that is, for any point \(y \in \mathcal{P}_{AL}\), there exists a point \((x', y') \in \mathcal{P}_{CF}\) with the same objective value; particularly, \(z_{\ast_{AL}} \geq z_{\ast_{CF}}\). Moreover, there are instances where this dominance is strict.

3.2 The Set Partitioning q-Route Formulation

We now turn our attention to a formulation based on q-routes [11]. A q-route is similar to a route, presented in the CmVRP definition, with the difference that it allows the existence of cycles, that is, replacing the condition \(v_i \neq v_j, \forall i \neq j\) with the condition \(v_i \neq v_{i+1}, vi = 1, \ldots, l - 1\). We still enforce that the total demand of a route is at most \(Q\) and, in the case of a q-route, the demand of each customer is accumulated as many times as it is visited. The set of all q-routes is denoted as \(\Omega\).
In this way, an integer solution for the CmVRP needs to visit each customer \( w \in N \) only once by traversing some routes among all in \( \Omega \). This can be modeled as a Set Partitioning Problem constrained to choose exactly \( K \) routes. Given the decision variables

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

the CmVRP Set Partitioning \( q \)-Route formulation (\( qR \)) is as follows:

\[
\begin{align*}
(qR) \quad \min & \quad \sum_{r \in \Omega} c_r \lambda_r \\
\text{s.t.} & \quad \sum_{r \in \Omega} \lambda_r = K, \quad (2a) \\
& \quad \sum_{r \in \Omega} h_{wr} \lambda_r = 1, \quad \forall w \in N, \quad (2c) \\
& \quad 0 \leq \lambda_r \leq 1, \quad \forall r \in \Omega, \quad (2d) \\
& \quad \lambda_r \in \mathbb{Z}, \quad \forall r \in \Omega. \quad (2e)
\end{align*}
\]

The \( h_{wr} \) coefficients represent the number of times a customer \( w \in N \) is visited by a \( q \)-route \( r \in \Omega \). The cost of a \( q \)-route \( r \) is given by \( c_r \), and the number of \( q \)-routes that can be traversed is constrained by (2b). Constraints (2c) impose that all customers must be visited exactly once. This model was introduced for the CVRP in Balinski & Quandt [6] and is used in the CmVRP by Fukasawa et al. [25].

As the number of possible \( q \)-routes is exponential in the size of the graph, the solution of the LP relaxation of \( qR \) is done using column generation [17, 7], where variables are added to a master LP as needed by solving a pricing subproblem. The use of \( q \)-routes instead of routes in the \( qR \) is precisely justified by this method: the pricing subproblem of \( q \)-routes is weakly NP-hard, while pricing routes is strongly NP-hard. We also note that much stronger set partitioning formulations exist where additional constraints are imposed in \( q \)-routes. However, for the sake of simplicity, we restrict ourselves to only set partitioning formulations using \( q \)-routes, but our results can be easily adapted to these other stronger formulations.

Next, we present Theorem 2, that tightly links the AL and the \( qR \) formulations. Note that if a formulation \( F \) dominates a formulation \( G \) and vice-versa, we say that \( F \) and \( G \) are equivalent.

**Theorem 2** (Fukasawa et al. [25]) The formulations AL and \( qR \) are equivalent, that is, for any point \( \lambda \in P_{AL} \), there exists a point \( \lambda^* \in P_{qR} \) with the same objective value and vice-versa; particularly, \( \hat{z}^*_{AL} = \hat{z}^*_{qR} \).

### 3.3 The Arc-Item Formulation

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

\[
x^i_a = \begin{cases} 
1 & \text{if item } i \text{ is carried over arc } a \\
0 & \text{otherwise}
\end{cases}, \quad \forall a \in A, i \in V.
\]

The formulation is:

\[
(AI) \quad \min \sum_{a \in A} \sum_{i \in V} \tilde{d}_{ai} q^i x^i_a \quad (3a)
\]

\[
\text{s.t.} \quad \sum_{a \in \delta^+(0)} x^0_a = K, \quad (3b)
\]

\[
\sum_{a \in \delta^-(w)} x^0_a = 1, \quad \forall w \in N, \quad (3c)
\]

\[
\sum_{a \in \delta^+(w)} x^0_a = 1, \quad \forall w \in N, \quad (3d)
\]

\[
\sum_{a \in \delta^-(t)} x^t_a = 0, \quad \forall t \in N, \quad (3e)
\]
We start by providing an example that illustrates what has been said before, that item 0 is always visited before any arcs in all routes and, therefore, variables $x^0_a$ just merely indicate if an arc $a$ is present in some route or not. The objective function (3a) minimizes the total cost of all the selected arc-items and, by considering $q^i$ as the curb weight, the objective function correctly considers the total weight (load plus curb weight) picked up prior to traversing $a$.

The selected arc-items whose items are indexed by 0 represent a full route in the classical sense, which also plays the role of a guide route: if an arc-item with item other than 0 is selected in an arc, then the arc-item with item 0 must also be selected in the same arc. With this in mind, we have that Constraint (3b) makes the depot out-degree equal the number of vehicles, while constraints (3c-3d) enforce that each customer has, with respect to item 0, an in-degree and out-degree of one. Note that these constraints play the same role that constraints (1b-1d) do for the AL model.

Constraints (3e) prevent each item of entering its customer of origin, and constraints (3f) enforce that, in fact, each customer’s item leaves its origin. Constraints (3g) state that all the items entering a given customer $w$, which are not the item from $w$, must also leave it. The aim of constraints (3h) is to tie up the arc-items under the guide arc-item 0, i.e., items can pass in an arc only if a vehicle is passing there. Constraints (3i) enforce that the total cumulative demand in each route does not exceed the vehicle capacity.

From our experimental results, we state Theorem 3 that shows how AI theoretically compares to other formulations.

**Theorem 3** The formulation AI does not dominate AL, qR nor CF; conversely, neither of these dominate AI.

As a final note, we remark that the CmVRP AI formulation is very similar to other formulations to the CVRP found in the literature using variables that have the same purpose, e.g., the MCF1a (Letchford & Salazar-González [42]) and the MCF2a (Letchford & Salazar-González [42, 43]). Beyond the objective function, the key difference is how capacity constraints are enforced, which is done in AI via constraints (3i), and the fact that some of the other formulations cited above are strengthened by adding more variables and/or constraints. The reason we chose to use formulation AI in this work, even though it is likely weaker than other similar formulations in the literature, is due to its simplicity and the fact that it suffices to capture aspects that are complementary to the AL formulation, as will be shown in the next section. Indeed, Letchford & Salazar-González [43] already mention that MCF2a is somewhat expensive, so we decided that attempting to use these stronger formulations would not be beneficial.

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

### 4.1 Why combine?

We start by providing an example that illustrates what has been said before, that AI and AL capture complementary aspects of the problem. Intuitively, while the AI formulation captures the precedence relationship (having visited a vertex before going through an arc), the AL captures the load relationship (how much load is carried through an arc). The complementary aspect of these two relationships can be seen by analyzing the literature. For instance, the improvements in some of the multi-commodity-flow formulations that are similar to AI are made via inequalities that better represent the load carried by a vehicle in an arc or vertex [42]. On the other hand, some of the improvements in the AL formulation [25] come from coupling the arc-load variables with $q$-routes to better capture the precedence relationship.

The main idea of formulation AIL (presented in Subsection 4.2) is to make sure the load carried in an arc is consistent according to both formulations. Since the load carried in an arc $a \in A$ in the AL formulation is $\sum_{\ell \in \mathcal{L}} f^\ell_a$ and in the AI formulation is $\sum_{i \in \mathcal{I}} q^i f^i_a$, the formulation makes sure these quantities are the same. In Figure 1 we present optimal solutions for a small instance S-n05-k1 ($n = 5$, $K = 1$, and $Q = 15$ with $q^0 = 2.25$) modeled with the AL, AI, and AIL relaxed formulations. We start by looking at the AL solution in Figure 1a. Note that as the AL and the qR formulations are
equivalent (see Theorem 2), the solution to AL can be written as a combination of (fractional) $q$-routes. The fractional solution presented in Figure 1a can be written as the sum of three fractional $q$-routes, where the decision variables related to the $q$-routes all have value $1/3$. Also note that these $q$-routes contain subtours. This allows us to see how combining the AL formulation with the AI formulation may be beneficial, since the AI formulation does not permit visiting a vertex $i$ having already visited it before, which would happen if one follows the $q$-routes.

![Figure 1](image_url)

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

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

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

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

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

It is not hard to extend the above analysis of Figure 1b to show that this solution cannot be represented using AL variables, thus also showing the value of hybridizing the two formulations. Indeed, Figure 1c presents an optimal solution of the LP relaxation for the instance S-n05-k1 using the hybridized AIL formulation (presented next, in Subsection 4.2), and, as the solution is integer, it is also an optimal solution to the integer version of the formulation. An optimal solution to the hybridized formulation is to use $q$-route $\langle 0, d, c, b, a, 0 \rangle$, which is consistent with both AI and AL formulations.

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

As was previously discussed, the AI formulation is aware of the items carried over each arc via arc-items, and the AL has control over the accumulated load weight in each arc via arc-loads. The idea behind the Arc-Item coupled to the Arc-Load formulation, also referred to as the Arc-Item-Load formulation (or AIL), is to combine those two features. The AIL formulation consists of the entire AI formulation (3a-3k), all the constraints of the AL (1b-1j), and
the following additional constraints:

\[ \begin{align*}
    x^0_a &= \sum_{\ell \in L} y^\ell_a, \quad \forall a \in A, \\
    \sum_{i \in N} q^i x^i_a &= \sum_{\ell \in L} q^\ell y^\ell_a, \quad \forall a \in A.
\end{align*} \tag{4a} \tag{4b} \]

The coupling constraints \((4a)\) correspond to the classical CVRP constraints that state that an arc is used (and therefore used carrying item 0) if and only if it is used carrying a load in \(L\). As previously mentioned, constraints \((4b)\) ensure that the load carried along arc \(a\) is consistent between the two formulations.

It is easy to see that constraints \((1b-1d)\) are redundant, and thus can be removed. Thus, the AIL formulation has the variables \([x^i_a]\) and \([y^\ell_a]\), the objective function \((3a)\), and the constraints \((3b-3k)\), \((1e-1j)\), and \((4a-4b)\). Furthermore, we have the following theorem about the AIL formulation and its building blocks.

**Theorem 4** The formulation AIL dominates AI and AL, that is, for any point \((x, y)\) in \(\mathcal{P}_{\text{AIL}}\), there exist points \(x \in \mathcal{P}_\text{AI}\) and \(y \in \mathcal{P}_\text{AL}\) with the same objective value; particularly, \(z^*_\text{AIL} \geq z^*_\text{AI}\) and \(z^*_\text{AIL} \geq z^*_\text{AL}\). Moreover, there are instances where these dominance relations are strict.

**Proof.** By construction, AIL dominates AI.

Let \((x, y)\) be any solution to the relaxation of the AIL formulation. By construction, we have that \(y\) satisfies all the constraints of AL. We now use the coupling constraints \((4a-4b)\) to substitute the variables \([x^i_a]\) by \([y^\ell_a]\) in AIL’s objective function \((3a)\), obtaining

\[\sum_{a \in A} \sum_{i \in V} \hat{h}^i_a \lambda_r = \sum_{a \in A} \sum_{i \in N} q^i x^i_a = \sum_{a \in A} \sum_{i \in N} q^i y^i_a,\]

which matches the AL’s objective function \((1a)\), guaranteeing that \(z^*_\text{AIL} \geq z^*_\text{AL}\). Thus, we have that AIL dominates AL.

By Theorem 3, AI does not dominate AL and neither does AL dominate AI, completing the proof.

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

There is a different alternative to combine the AI formulation with the AL formulation. As mentioned before, Theorem 2 says that the qR formulation is equivalent to the AL. Therefore, we attempted to combine the qR and AI formulations, and this is what this subsection is devoted to.

The new model comprises the entire AI model \((3a-3k)\), all the constraints of the qR model \((2b-2e)\), and the coupling constraints

\[ x^i_a = \sum_{r \in \Omega} \hat{h}^i_a \lambda_r, \quad \forall a \in A, i \in V, \tag{5} \]

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

Using the coupling constraints \((5)\), we can simplify the new coupled model. It is not hard to see that Constraint \((3b)\) implies Constraint \((2b)\) and also that constraints \((3f)\) imply constraints \((2c)\). Therefore, we can remove the constraints \((2b-2c)\) from the AILqR model. The resulting formulation is in the Dantzig-Wolfe explicit master form [53]. The following two results are easy to see.

**Theorem 5** The formulation AILqR dominates qR, that is, for any point \(x \in \mathcal{P}_{\text{AILqR}}\), there exists a point \(\lambda \in \mathcal{P}_{\text{qR}}\) with the same objective value; particularly, \(z^*_\text{AILqR} \geq z^*_\text{qR}\). Moreover, there are instances where this dominance is strict.

**Proof.** Let \((x, \lambda)\) be any solution to the relaxation of the AILqR formulation. By the simplifications we mentioned above, we know that \(\lambda\) immediately satisfies the constraints of the qR model. Now we use the coupling constraints \((5)\) to substitute the \([x^i_a]\) variables in the objective function \((3a)\) of AILqR, resulting in

\[\sum_{a \in A} \sum_{i \in V} \hat{h}^i_a \lambda_r = \sum_{r \in \Omega} \left( \sum_{a \in A} \sum_{i \in N} q^i \hat{h}^i_a \right) \lambda_r = \sum_{r \in \Omega} c_r \lambda_r,\]
matching the objective function (2a) of the \( q R \). Thus, we have that \( z_{AlqR}^* \geq z_{qR}^* \). Our computational experiments show that there are instances where \( z_{AlqR}^* > z_{qR}^* \), therefore, the main result follows. \( \square \)

**Theorem 6** The formulation \( AlqR \) dominates \( AI \), that is, for any point \( x \in P_{AlqR} \), there exists a point \( x \in P_{Al} \) with the same objective value; particularly, \( z_{AlqR}^* \geq z_{AI}^* \). Moreover, there are instances where this dominance is strict.

**Proof.** The dominance of the \( AlqR \) over the \( AI \) is guaranteed by the former construction. By Theorem 3, the \( AI \) does not dominate \( q R \), which finishes the proof. \( \square \)

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

Let \( \varphi_{\alpha}^d \), for all \( \alpha \in A, i \in V \), be the dual variables associated with the coupling constraints (5). The reduced cost \( \tilde{c}_r \), of a \( q \)-route/column \( r \in \Omega \), can be calculated as

\[
\tilde{c}_r = c_r - \sum_{a \in A} \sum_{i \in V} (-h_{a,i}) \varphi_{\alpha}^d = c_r + \sum_{a \in A} \sum_{i \in V} \varphi_{\alpha}^d h_{a,i} = \sum_{a \in A} \sum_{i \in V} \varphi_{\alpha}^d h_{a,i}.
\]

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

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

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

We prove in Theorem 7 that Prc-AlqR is strongly NP-hard by using the directed Hamiltonian cycle problem.

**Theorem 7** The Prc-AlqR is strongly NP-hard. \( \square \)

**Proof.** We prove the result by showing that Prc-AlqR can be used to solve the directed Hamiltonian cycle problem.

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

The input data for Prc-AlqR would, therefore, be the complete digraph \( D' = (V, A') \), capacity \( Q = n - 1 \), and \( q^t = 1 \) for all \( t \in N \). The costs \( a_{\alpha}^i \) would be as follows:

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

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

Therefore, the minimum cost solution \( r^* \) to Prc-AlqR has negative value if and only if \( r^* \) is a cycle using only arcs of \( A \). Thus, it is easy to see that \( D \) has a directed Hamiltonian cycle if and only if \( r^* \) has value \(-n\). \( \square \)

We note that the hardness proof relies on the fact that each arc-item combination has a different cost, which in turn comes from the fact that we have one coupling constraint (5) for each arc-item combination. Thus, to define a more tractable pricing problem, a more “loosely coupled” formulation becomes necessary. We refrain from defining precisely what loosely coupled means, and proceed to actually presenting the formulation in question.
4.4 The Arc-Item-Load Coupled to the Set Partitioning q-Route Formulation

To construct this new formulation, referred to as AILqR, we start from the entire simplified AIL formulation (3a-3k, 1e-1j, 4a-4b) and the qR constraints (2b-2e). We couple these together by making use of the following relationship between the arc-loads \( |y^\ell_a^f| \) and the q-routes:

\[
y^\ell_a^f = \sum_{r \in \Omega} \hat{h}^\ell_{ar} \lambda_r, \quad \forall a \in A, \ell \in L,
\]

where the \( \hat{h}^\ell_{ar} \) coefficient represents the number of times a load \( \ell \in L \) is carried over an arc \( a \in A \) in a q-route \( r \in \Omega \).

Theorem 2 states that the AIL and the qR formulations are equivalent, and thus we remove the AIL constraints (1e-1h).

We now prove that constraints (2b-2c) from qR are implied by other constraints that are left in the model, and therefore can also be eliminated. Starting from (3b), applying (4a), and finally using (6), (2b) can be derived as follows:

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

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

Similarly, using (3d), (4a) and (6), constraint (2c) can be derived, for all \( w \in N \), as:

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

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

In addition, using (6), we replace (4a-4b) with the following:

\[
x^0_a = \sum_{\ell \in L} \sum_{r \in \Omega} \hat{h}^\ell_{ar} \lambda_r, \quad \forall a \in A,
\]

\[
q^\ell x^\ell_a = \sum_{\ell \in L} \sum_{r \in \Omega} \hat{h}^\ell_{ar} \lambda_r, \quad \forall a \in A.
\]

Finally, we remove the \( |y^\ell_a^f| \) variables, as well as any other constraints (e.g. nonnegativity) involving them. Note that the model still implicitly uses the concept of arc-load to make the connection between the arc-items and the q-routes.

Summing up, the AILqR variables are the \( |x^\ell_a^f| \) and \( \lambda_r \) and the formulation consists of the objective function (3a) and the constraints (3b-3k), (2d-2e), and (7a-7b).

The construction of the AILqR formulation implies the following theorem.

\textbf{Theorem 8} The formulations AIL and AILqR are equivalent, that is, for any point \((x, y) \in P_{AIL}\), there exists a point \((x, y, \lambda) \in P_{AILqR}\) with the same objective value and vice-versa; particularly, \( z_{AIL}^* = z_{AILqR}^* \).

We now show that this new way of coupling the formulations has a more tractable pricing problem. Let \( \pi_a \) and \( \mu_a \), for all \( a \in A \), be the dual variables of the constraints in (7a) and (7b), respectively. Therefore, the reduced cost \( \bar{c}_r \), of a column \( r \in \Omega \), can be calculated as given below.

\[
\bar{c}_r = c_r - \left( \sum_{a \in A} \left( -\hat{h}^\ell_{ar} \right) \pi_a + \sum_{a \in A} \left( \hat{h}^\ell_{ar} \right) \mu_a \right) = c_r + \sum_{a \in A} \sum_{\ell \in L} \left( \pi_a + \mu_a \ell \right) \hat{h}^\ell_{ar} = \sum_{a \in A} \sum_{\ell \in L} \left( \pi_a + \mu_a \ell \right) \hat{h}^\ell_{ar}.
\]

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

Finding the minimum reduced cost q-route can be solved as a Shortest Path Problem with Resource Constraints (SPPRC) [34, 54] that minimizes \( \bar{c}_r \) while restricting the accumulated load to respect the vehicle capacity \( Q \). This version of the problem, referred to as Pac-AILqR, is weakly NP-hard.

4.5 The Arc-Item Coupled to the Set Partitioning t-Route Formulation

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

\[
\Omega = \bigcup_{t \in V} \Psi_t.
\]

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

\[
(tR) \quad \min \sum_{t \in V} \sum_{r \in \Psi_t} c_r \lambda_r \quad \text{s.t.} \quad
\begin{align*}
\sum_{r \in \Psi_t} \lambda_r &= K, \\
\sum_{r \in \Psi_t} \lambda_r &= 1, \quad \forall t \in N, \\
\sum_{r \in \Psi_t} \bar{h}_{ar} \lambda_r &\leq \sum_{s \in \Psi_0} \bar{h}_{as} \lambda_s, \quad \forall a \in A, \ t \in N, \\
0 &\leq \lambda_r \leq 1, \quad \forall r \in \Omega, \\
\lambda_r &\in \mathbb{Z}, \quad \forall r \in \Omega.
\end{align*}
\]

Comparing to the \( qR \) formulation, the objective function (8a) explicitly shows that the number and the role of the \( t \)-routes differs from those of the \( q \)-routes. The depot out-degree is now modeled only by the number of \( t \)-routes that start at it, as stated by the Constraint (8b). Constraints (8c) ensure exactly one \( t \)-route is selected for each customer. Constraints (8d) ensure that \( t \)-routes must follow the ones of the depot.

We then propose the AI coupled to the \( tR \) formulation (\( AItR \)). It comprises the entire AI model (3a-3k), all the constraints of the \( tR \) (8b-8f), and the coupling constraints

\[
x'_ia = \sum_{r \in \Psi_t} \bar{h}_{ar} \lambda_r, \quad \forall a \in A, t \in V.
\]

It is not hard to show that the coupling constraints (9) can be used to obtain a simplified version of the model. From constraint (3b), the depot out-degree constraint (8b) can be obtained. Constraints (3f) imply the constraints (8c) whereas constraints (3h) imply the ones in (8d). Furthermore, we present the following theorems.

**Theorem 9** The formulation \( AItR \) dominates AI, that is, for any point \( (x, \lambda) \in \mathcal{P}_{AItR} \), there exists a point \( x \in \mathcal{P}_{AI} \) with the same objective value; particularly, \( z^*_{AItR} \geq z^*_{AI} \). Moreover, there are instances where this dominance is strict.

**Proof.** The result follows by the construction of the formulation \( AItR \) and experimental results.

We can draw the following theorems from our experimental results.

**Theorem 10** The formulation \( AItR \) does not dominate nor is dominated by \( AL \).

**Theorem 11** The formulation \( AItR \) does not dominate \( AIL \).

We now focus on the pricing problem of \( AItR \). Note that in each column generation round, \( n \) pricing subproblems will be run (one for each item). Let \( \eta'_a \), for all \( a \in A, t \in V \), be the dual variables associated to the coupling constraints (9). Thus, the reduced cost \( \bar{c}_r \) of a column \( r \in \Psi_t \) can be calculated as

\[
\bar{c}_r = c_r - \sum_{a \in A} (-\bar{h}_{ar}) \eta'_a = c_r + \sum_{a \in A} \eta'_a \bar{h}_{ar} = \sum_{a \in A} \eta'_a \bar{h}_{ar}.
\]

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

A pricing subproblem for a specific item \( t \) can be treated as a SPPRC that minimizes \( \bar{c}_r \) subject to the total accumulated load not exceeding the vehicle capacity \( Q \). In each round of column generation, we have to solve this pricing subproblem.
4.6 Overview of the Formulations

We finish this section summarizing the relationship between the formulations of interest in Figure 2. We highlight that AL and qR are equivalent, and, among the proposed formulations, AL and ALqR are also equivalent. In this situation, the relations of one automatically apply to the other. We believe that ALqR is the strongest formulation in this paper. However, there is no formal proof that it dominates AL/AILqR and AlIR, and moreover its pricing subproblem is strongly NP-hard. The relaxations of the other formulations presented are either compact or have a weakly NP-hard pricing problem.

The pair AIL/AILqR provides a feasible compromise between theoretical quality and computational hardness, as they probably dominate every formulation in Figure 2, except possibly AlqR. The LP relaxations of either AI and AlIR are incomparable with the LP relaxations of the pair AIL/qR. The AI is incomparable even with the CF. By incomparable, we mean that neither one dominates the other.

5 Computational Experiments

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

5.1 Instances

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

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

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

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

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

5.3 Experiments on Small Instances

Table 1 presents the results of experiments made on small instances to compare the AL formulations, we use CPLEX in the deterministic mode and with just one thread. On the other hand, the column generation formulations are solved with CPLEX in opportunistic mode with four threads. The experiments were run on a computer with four Intel Xeon Gold 6142 @ 2.60GHz CPU chips and 252GiB of RAM. The machine has 64 physical cores that can run at most one thread each and the operating system is the GNU/Linux Ubuntu 18.04. The number of test threads we run in parallel is at most the number of cores minus one.

Table 1 presents the results of experiments made on small instances to compare the AL formulations, we use CPLEX in the deterministic mode and with just one thread. On the other hand, the column generation formulations are solved with CPLEX in opportunistic mode with four threads. The experiments were run on a computer with four Intel Xeon Gold 6142 @ 2.60GHz CPU chips and 252GiB of RAM. The machine has 64 physical cores that can run at most one thread each and the operating system is the GNU/Linux Ubuntu 18.04. The number of test threads we run in parallel is at most the number of cores minus one.

Table 1. Results of LP relaxations of AL, AI, and AIL on small instances

<table>
<thead>
<tr>
<th>Inst</th>
<th>Q</th>
<th>#</th>
<th>T(s)</th>
<th>(\hat{\xi})</th>
<th>#I</th>
<th>G(%)</th>
<th>T(s)</th>
<th>(\hat{\xi})</th>
<th>#I</th>
<th>G(%)</th>
<th>T(s)</th>
<th>(\hat{\xi})</th>
<th>#I</th>
<th>G(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-n04-k1</td>
<td>10</td>
<td>0.0</td>
<td>112.50</td>
<td>0</td>
<td>0</td>
<td>112.50</td>
<td>0</td>
<td>112.50</td>
<td>0</td>
<td>0.0</td>
<td>112.50</td>
<td>0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>S-n05-k1</td>
<td>15</td>
<td>0.0</td>
<td>233.25</td>
<td>1.1</td>
<td>0</td>
<td>229.08</td>
<td>2.8</td>
<td>0.0</td>
<td>235.75</td>
<td>0</td>
<td>0.0</td>
<td>235.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n08-k2</td>
<td>10</td>
<td>0.0</td>
<td>2,599.00</td>
<td>9.1</td>
<td>0</td>
<td>2,859.00</td>
<td>0</td>
<td>0.0</td>
<td>2,859.00</td>
<td>0</td>
<td>0.0</td>
<td>2,859.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n09-k3</td>
<td>15</td>
<td>0.0</td>
<td>1,078.25</td>
<td>0.2</td>
<td>0</td>
<td>1,074.08</td>
<td>0.6</td>
<td>0.0</td>
<td>1,080.75</td>
<td>0</td>
<td>0.0</td>
<td>1,080.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n09-k3-d</td>
<td>15</td>
<td>0.0</td>
<td>339.52</td>
<td>8.2</td>
<td>0</td>
<td>360.95</td>
<td>2.4</td>
<td>0.0</td>
<td>369.75</td>
<td>0</td>
<td>0.0</td>
<td>369.75</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n10-k3</td>
<td>50</td>
<td>0.0</td>
<td>13,034.45</td>
<td>4.2</td>
<td>0</td>
<td>13,483.67</td>
<td>0.9</td>
<td>0.1</td>
<td>13,576.25</td>
<td>0.2</td>
<td>0.1</td>
<td>13,576.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n13-k4</td>
<td>50</td>
<td>0.0</td>
<td>908.25</td>
<td>1.7</td>
<td>0</td>
<td>923.50</td>
<td>0</td>
<td>0.3</td>
<td>923.50</td>
<td>0</td>
<td>0.3</td>
<td>923.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-n25-k5</td>
<td>80</td>
<td>0.3</td>
<td>36,476.00</td>
<td>2</td>
<td>0.7</td>
<td>37,053.88</td>
<td>0.5</td>
<td>84.7</td>
<td>37,059.15</td>
<td>0.5</td>
<td>84.7</td>
<td>37,059.15</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The results for the instances S-n05-k1 and S-n08-k2 are enough to imply that AI does not dominate nor is dominated by AL, thus proving part of Theorem 3. Moreover, from the numbers in Table 1, we see that, in instance class S, AIL has an average gap of 0.1%, while AL and AI have average gaps of 0.9% and 3.3%, respectively. Therefore, while AL and AI do not dominate each other, AL performed better than AI. In addition, we point out that, in this instance class, the different structures of AI and AL combined into AIL are able to produce better bounds than each of them individually. However, the major concern is the time spent to solve the LP relaxation of the larger instance S-n25-k5 with AIL.

We also performed experiments using instance class S and the LP relaxation of AIL\textsubscript{qR}. Table 2 shows the corresponding results. As this model relies on column generation, the table presents some of that relevant data under the group “CG Stats”. The number of column generation rounds is reported in column “# Rnds” and the total number of generated columns is given by “# Gen Cols”. The next two columns present the total time spent generating columns (“CG T(s)” and solving the LP master problem (“LP T(s)”)). The remaining columns are the same as in Table 1. The last row once more presents the total number of instances and average values.

We highlight, from tables 1 and 2, the expected equivalence of \(\hat{\xi}\) and gap values between AIL\textsubscript{qR} and AIL, as expected due to Theorem 8. We highlight the extremely high time to solve the LP relaxation of AIL\textsubscript{qR} for instance S-n25-k5. It takes a total of 4,428.5s, 201.2s due to the CG steps and 4,223s due to solving the LP master, which means that only 4.5% of the time is used to generate the 40,614 columns through the rounds, and 95.4% of the time is spent solving the 1,473 linear programs. In this instance, each linear program takes an average of 2.9s to be solved. While not much time was spent generating the columns, the high number of rounds implies that a large number of linear programs need to be solved. We note that the total time of solving just the LP relaxation of AIL\textsubscript{qR} for S-n25-k5 (4,428.5s) is extremely high whether compared to solving the LP relaxation of AIL (84.7s) or even compared to running the exact integer BCP itself (2.9s).

In fact, if one is willing to deal with this type of pricing subproblem, a reasonable way is to directly use a specialized relaxation of the problem, as by Theorem 1, it is dominated by another relaxed model. However, the solution is fractional. From the results in Table 3, one can state that 97.2% of the columns in this test are generated by the first third of column generation iterations.

Another interesting note is that the LP relaxation of \( \text{AIL}_R \) on small instances. When comparing to the average number of rounds of \( \text{AIL}_R \) (216.8), the same statistic in \( \text{AIL}_I \) (185.2) is similar, albeit somewhat smaller. However, the number of generated columns is about seven times larger than in the former model, the CG time is about 40 times larger, and the LP time and the total time are about 60 times larger. The main justification of \( \text{AIL}_I \) is that it has a pseudo-polynomial time pricing subproblem, such as \( \text{AIL}_R \). However, even though the number of column generation rounds is almost always smaller, the consistently higher number of generated columns led to larger linear programs. Each linear program took an average of 214.8s for the S-n25-k5 instance, against the mean of 2.9s for the \( \text{AIL}_R \). This gives us some evidence that decreasing the number of rounds may not be good if accompanied by a big increase in the number of generated columns.

These average numbers are mainly influenced by the test over the instance S-n25-k5. For the sake of completeness, we report some statistics about a test of S-n25-k5 limited to 86,400s, a more reasonable time limit: it performed 424 rounds generating 296,411 columns, the CG procedure steps took 2,174.6s while the LP solver steps consumed 84,192s. Note that this suggests that 97.2% of the columns in this test are generated by the first third of column generation iterations.

Another interesting note is that the LP relaxation of \( \text{AIL}_R \) instance S-n04-k1 reaches the same value of \( \frac{2}{n} = 112.5 \) as the other relaxed models tested, however the solution is fractional. From the results in Table 3, one can state that \( \text{AIL}_R \) does not dominate the pair \( \text{AIL}_L_R \) and \( \text{AIL} \) (implying Theorem 11). These results also imply that \( \text{AIL}_R \) does not dominate nor is dominated by \( \text{AL} \) (implying Theorem 10), as well as implying that \( \text{AL} \) does not dominate \( \text{AIL}_R \) (proving part of Theorem 9).

This paper did not present experimental results over the CF model, as by Theorem 1, it is dominated by AL. Moreover, there are also no experiments using \( \text{AL}_R \) due to the fact that solving its relaxation is strongly NP-hard, as stated by Theorem 7. In fact, if one is willing to deal with this type of pricing subproblem, a reasonable way is to directly use a specialized

<table>
<thead>
<tr>
<th>Inst</th>
<th>Q</th>
<th># Rnds</th>
<th># Gen Cols</th>
<th>CG T(s)</th>
<th>LP T(s)</th>
<th>T(s)</th>
<th>( \zeta )</th>
<th>#I</th>
<th>G(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-n04-k1</td>
<td>10</td>
<td>6</td>
<td>30</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>112.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n05-k1</td>
<td>15</td>
<td>16</td>
<td>128</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>235.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n08-k2</td>
<td>10</td>
<td>79</td>
<td>169</td>
<td>0.0</td>
<td>0.5</td>
<td>0.5</td>
<td>2,859.00</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n09-k3</td>
<td>15</td>
<td>35</td>
<td>283</td>
<td>0.0</td>
<td>0.5</td>
<td>0.6</td>
<td>1,080.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n09-k3-d</td>
<td>15</td>
<td>45</td>
<td>321</td>
<td>0.0</td>
<td>0.5</td>
<td>0.5</td>
<td>369.75</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n10-k3</td>
<td>50</td>
<td>32</td>
<td>467</td>
<td>0.1</td>
<td>0.9</td>
<td>1.0</td>
<td>13,576.25</td>
<td>0.2</td>
<td>0</td>
</tr>
<tr>
<td>S-n13-k4</td>
<td>50</td>
<td>48</td>
<td>924</td>
<td>0.2</td>
<td>2.7</td>
<td>2.9</td>
<td>923.50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n25-k5</td>
<td>80</td>
<td>1,473</td>
<td>40,614</td>
<td>201.2</td>
<td>4,223.0</td>
<td>4,428.5</td>
<td>37,059.15</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>8</td>
<td>216.8</td>
<td>5,367</td>
<td>25.2</td>
<td>528.5</td>
<td>554.2</td>
<td>6</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2. Results of LP relaxation of \( \text{AIL}_R \) on small instances

<table>
<thead>
<tr>
<th>Inst</th>
<th>Q</th>
<th># Rnds</th>
<th># Gen Cols</th>
<th>CG T(s)</th>
<th>LP T(s)</th>
<th>T(s)</th>
<th>( \zeta )</th>
<th>#I</th>
<th>G(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-n04-k1</td>
<td>10</td>
<td>5</td>
<td>94</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>112.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>S-n05-k1</td>
<td>15</td>
<td>8</td>
<td>265</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>233.25</td>
<td>1.1</td>
<td></td>
</tr>
<tr>
<td>S-n08-k2</td>
<td>10</td>
<td>74</td>
<td>788</td>
<td>0.1</td>
<td>0.1</td>
<td>0.2</td>
<td>2,657.89</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>S-n09-k3</td>
<td>15</td>
<td>24</td>
<td>1,260</td>
<td>0.2</td>
<td>0.3</td>
<td>0.3</td>
<td>1,578.25</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>S-n09-k3-d</td>
<td>15</td>
<td>36</td>
<td>1,194</td>
<td>0.2</td>
<td>0.3</td>
<td>0.3</td>
<td>353.40</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td>S-n10-k3</td>
<td>50</td>
<td>24</td>
<td>2,591</td>
<td>0.3</td>
<td>0.9</td>
<td>1.3</td>
<td>13,282.60</td>
<td>2.3</td>
<td></td>
</tr>
<tr>
<td>S-n13-k4</td>
<td>50</td>
<td>63</td>
<td>5,612</td>
<td>1.9</td>
<td>5.4</td>
<td>7.3</td>
<td>910.50</td>
<td>1.4</td>
<td></td>
</tr>
<tr>
<td>S-n25-k5</td>
<td>80</td>
<td>1,248</td>
<td>304,946</td>
<td>7,728.1</td>
<td>268,072.0</td>
<td>275,863.0</td>
<td>36,749.63</td>
<td>1.3</td>
<td></td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>8</td>
<td>185.2</td>
<td>39,593.8</td>
<td>966.3</td>
<td>33,509.8</td>
<td>34,484.0</td>
<td>0</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 3. Results of LP relaxation of \( \text{AIL}_I \) on small instances

These average numbers are mainly influenced by the test over the instance S-n25-k5. For the sake of completeness, we report some statistics about a test of S-n25-k5 limited to 86,400s, a more reasonable time limit: it performed 424 rounds generating 296,411 columns, the CG procedure steps took 2,174.6s while the LP solver steps consumed 84,192s. Note that this suggests that 97.2% of the columns in this test are generated by the first third of column generation iterations.
method for the strongly NP-hard Elementary SPPRC [19], and practical experience indicates that several implementation improvements must be made, however that is not the focus of our paper.

### 5.4 Experiments on Regular Instances

Experimental results with regular benchmark instances from the literature have their gaps presented in Table 4 and their computational times in Table 5. These experiments were made over the 100 regular instances, where we set the time limit to 86,400s (24 hours). The reported gaps of the models’ LP relaxations were calculated against the objective value of the best available integer solutions.

The integer solutions were obtained as follows. For the majority of the instances, we used the BCP program of Fukasawa et al. [25] with a time limit of 604,800s (7 days). Besides that, to deal with some instances where some technical issues happened with this BCP program, we relied on our program with the (integer version of the) AIL model and a time limit of 2,592,000s (30 days). The two instances that ended up with no available integer solution, namely E-n101-k8 and P-n101-k4, are among the largest instances of each class, which leaves us with 98 (out of 100) instances with integer solutions.

To report the main results, first, we kept just the tests carried out over instances that have integer optimal or integer best known solutions available. In addition, we present only those tests where optimal solutions were found for the LP relaxations of all three models, leaving us with 71 instances. We comment that all the LP relaxations of AI and AL were solved within the time limit on all instances. These 71 instances have up to 67 vertices, as can be seen in column $n'$ in Table 4, where $n'$ represents the largest instance size in each class that remains after the above mentioned filters were applied.

<table>
<thead>
<tr>
<th>Inst</th>
<th>#</th>
<th>$n'$</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
<th>#I</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
<th>#I</th>
<th>Min</th>
<th>Avg</th>
<th>Max</th>
<th>#I</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>19</td>
<td>60</td>
<td>2.7</td>
<td>7</td>
<td>12.4</td>
<td></td>
<td>1.1</td>
<td>3.6</td>
<td>10.1</td>
<td></td>
<td>0.2</td>
<td>1.9</td>
<td>5.7</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>15</td>
<td>67</td>
<td>3.9</td>
<td>9.1</td>
<td>22.2</td>
<td></td>
<td>1.3</td>
<td>6.7</td>
<td>18.9</td>
<td></td>
<td>1.1</td>
<td>4.3</td>
<td>17.2</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>4</td>
<td>31</td>
<td>5.4</td>
<td>10.2</td>
<td>19.6</td>
<td></td>
<td>0.7</td>
<td>4.3</td>
<td>14.1</td>
<td></td>
<td>0</td>
<td>3</td>
<td>11.4</td>
<td>1</td>
</tr>
<tr>
<td>P</td>
<td>20</td>
<td>65</td>
<td>0.5</td>
<td>9.5</td>
<td>17.9</td>
<td></td>
<td>0</td>
<td>3.2</td>
<td>8.5</td>
<td>1</td>
<td>0</td>
<td>1.8</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>V</td>
<td>13</td>
<td>48</td>
<td>0</td>
<td>9.5</td>
<td>22.9</td>
<td></td>
<td>1</td>
<td>4.4</td>
<td>12.8</td>
<td>2</td>
<td>0</td>
<td>2.5</td>
<td>9</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>67</td>
<td></td>
<td>8.8</td>
<td></td>
<td>1</td>
<td></td>
<td>4.3</td>
<td></td>
<td>3</td>
<td></td>
<td>2.6</td>
<td></td>
<td>4</td>
</tr>
</tbody>
</table>

The average gaps of the relaxations of AI, AL, and AIL, are, respectively, 8.8%, 4.3%, and 2.6%. We can see that the behavior presented in the small instances also happens here, i.e., this sequence of gaps is in decreasing order. Considering we have that AIL dominates AI and AL (see Theorem 4), the last value being smaller is expected. However, the consistent ratio between the AI gap and the others may be attributed to the fact that each formulation captures different and complementary aspects of the CmVRP, and putting them together tends to produce a stronger formulation that has been worth exploring.

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

In Figure 3, we also present a plot of the cumulative frequency of gaps in the instances. A point $(x, y)$ in this figure represents that $y$ instances have a gap at most $x\%$. One can see that the AIL relaxed formulation reaches a gap of at most 5% in 64 instances. For the other two formulations, AI and AL only 50 and 11 instances have a gap of at most 5%, respectively. Indeed, this chart shows that the average data contained in Table 4 are not biased by only a few instances.

Despite having presented a promising experimental gap, the main drawback of AIL is the computational time it requires to have its relaxation solved, as shown by Table 5. While AI and AL took an average of 16.8s and 19.6s in those 71 instances, the AIL computational time average is 17,409.7s (almost five hours), with some peaks of more than 77,400s (21.5 hours).
hours). In addition, we conducted experiments with our current simple implementation of the AIL$q_R$, and, the optimal relaxed solution was reached in only 15 instances within the same time limit of 86,400s.

Table 5. Computational times of the LP relaxations of AI, AL, and AIL

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

6 Concluding Remarks

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

We have also proposed the set partitioning AI$q_R$ and AIL$q_R$ formulations, that have pricing subproblems solvable in pseudo-polynomial time. While the former did not provide a computational advantage, important results were derived about the latter. We proved that AIL and AIL$q_R$ are equivalent, and besides, this relation was also important to provide an intuition about the good gap results of AIL, when compared to AI and AL. The advantage of the AIL may be attributed to the fact that each base formulation alone captures different and complementary aspects of the CmVRP, and putting them together tends to produce a stronger formulation that is worth exploring. The downside is the large amount of time spent in solving the tighter formulation.

We also contributed by proposing a somewhat natural way to try and combine AI with a set partitioning formulation, namely the AI$q_R$ formulation, which dominates the $q_R$ formulation, and thus, by transitivity, also dominates AL. However, such a natural formulation is not likely to be useful in practice since its pricing subproblem is strongly NP-hard.

The results of this paper point to multiple possibilities of research. The strength of the AIL can be further studied in order to discover its relation with the several families of cuts for the CVRP available in the literature, e.g., proving whether the cuts are useful over the model, including cuts which eliminate cycles of length two, which already exist for the AL. The experimental results concerning the gaps suggests one may benefit from using AIL in BB and BC methods, but further
research is needed to speed up its high computational cost. Considering the AIL and AILqR equivalence, speeding up the column generation process and also strengthening it by using other structures such as ng-routes [5] may make the latter viable to be used in a BP or BCP method. It is also worth searching for a formulation whose LP relaxation (i) can be solved in pseudo-polynomial time, and (ii) gives bounds that are better than the ones from AILqR.

Acknowledgments

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

References


A Full Experimental Results on Regular Instances with Relaxed Models

In Table 6 we present the full experimental results that are condensed in Tables 4 and 5 in Section 5. It is worth emphasizing that the original values (reported on Table 6) of vehicle capacity \( Q \) of some instances were preprocessed by being divided by a scaling factor, resulting in \( Q' \), as in the following:

- E-n13-k4: \( Q' = 6,000/100 = 60 \);
- E-n22-k4: \( Q' = 6,000/100 = 60 \);
- E-n30-k3: \( Q' = 4,500/25 = 180 \); and
- P-n22-k8: \( Q' = 3,000/100 = 30 \).

<table>
<thead>
<tr>
<th>Inst</th>
<th>( Q ) #</th>
<th>Arc-Item</th>
<th>Arc-Load</th>
<th>Arc-Item-Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-n32-k5 100</td>
<td>1.1</td>
<td>39,989.20</td>
<td>2.7</td>
<td>4.4</td>
</tr>
<tr>
<td>A-n33-k5 100</td>
<td>1.0</td>
<td>31,501.52</td>
<td>5.6</td>
<td>4.1</td>
</tr>
<tr>
<td>A-n33-k6 100</td>
<td>1.2</td>
<td>34,983.37</td>
<td>5.5</td>
<td>3.9</td>
</tr>
<tr>
<td>A-n34-k5 100</td>
<td>2.1</td>
<td>27,227.37</td>
<td>9.5</td>
<td>8.1</td>
</tr>
<tr>
<td>A-n36-k5 100</td>
<td>2.9</td>
<td>37,098.70</td>
<td>5.6</td>
<td>19.5</td>
</tr>
<tr>
<td>A-n37-k5 100</td>
<td>2.1</td>
<td>29,641.05</td>
<td>4.4</td>
<td>21.6</td>
</tr>
<tr>
<td>A-n37-k6 100</td>
<td>4.1</td>
<td>45,749.12</td>
<td>8.7</td>
<td>11.9</td>
</tr>
<tr>
<td>A-n38-k5 100</td>
<td>2.3</td>
<td>33,965.74</td>
<td>11.9</td>
<td>13.2</td>
</tr>
<tr>
<td>A-n39-k5 100</td>
<td>8.5</td>
<td>39,528.51</td>
<td>7.5</td>
<td>27.4</td>
</tr>
<tr>
<td>A-n39-k6 100</td>
<td>1.9</td>
<td>38,660.46</td>
<td>6.5</td>
<td>5.7</td>
</tr>
<tr>
<td>A-n44-k6 100</td>
<td>10.2</td>
<td>45,311.43</td>
<td>5.4</td>
<td>15.8</td>
</tr>
<tr>
<td>V</td>
<td>13</td>
<td>33.3</td>
<td>1</td>
<td>9.5</td>
</tr>
<tr>
<td>---</td>
<td>----</td>
<td>------</td>
<td>---</td>
<td>-----</td>
</tr>
<tr>
<td></td>
<td>71</td>
<td>16.8</td>
<td>1</td>
<td>8.8</td>
</tr>
</tbody>
</table>