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

Mauro Henrique Mulati[1,2,✉] Ricardo Fukasawa[3] Flávio Keidi Miyazawa[1]

mhmulati@ic.unicamp.br rfukasawa@uwaterloo.ca fkm@ic.unicamp.br

mhmulati@unicentro.br

[1]Institute of Computing, University of Campinas (UNICAMP), Campinas, SP, Brazil

[2]Department of Computer Science, Midwestern State University (UNICENTRO), Guarapuava, PR, Brazil

[3]Department of Combinatorics and Optimization, University of Waterloo, Waterloo, ON, Canada

Fri 3 Jun 2022 3:57pm -03:00

**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 Eglese & Bektaş [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 ($K$-TRP) [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 $K$-TRP 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 CVRP 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 $i$ if vertex $i$ has been visited prior to traversing the arc. It is worth noting that the arc-item formulation is very similar to multi-commodity-flow formulations with one commodity per vertex, that have been studied for the CVRP (see for instance Letchford & 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 CVRP.

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 MCF2b, based on a multi-commodity-flow idea. Beyond the binary variables $[x_a]$ indicating whether arc $a$ is used, their MCF2b formulation, whose main concept can be traced back to Garvin et al. [26], contains binary variables to indicate whether an arc $a$ is traversed by a vehicle before ($[f_a^i]$ variables) or after ($[g_a^i]$ variables) visiting customer $i$. This kind of formulation also has precedents in modeling the Traveling Salesman Problem (TSP) and some variations [40, 33]. The MCF1a formulation is conceptually similar to the MCF2b, with the difference that only the variables $[x_a]$ and $[f_a^i]$ are present. Moreover, the MCF2b model was strengthened by coupling it to a knapsack based formulation with exponentially many columns, each one corresponding to a packing pattern of the items. The pricing of the variables can be done in pseudo-polynomial time and the resulting formulation is called MCF2K.

Also based on the MCF2b, the Arc-Packing (AP) formulation couples the existing variables with a new set of arc-packing binary variables $[\psi_a^{ST}]$, which take the value of one if and only if a vehicle traverses arc $a$ after having visited the customers (packed) in $S$, and no more, and about to visit the ones (packed) in $T$, and no more. As $S, T \subseteq N$ (where $N$ is the set of all customers), the number of all possible columns is exponential, however its pricing subproblem can be solved in pseudo-polynomial time. In their work, Letchford & 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, MCF2b, and MCF1a; where the first dominates the second and so on. With a time limit of two hours (for the first four models), they carried out experiments including relaxations of these formulations over new CVRP instances with 16 vertices, divided into 12 families of 20 instances each, which helped to establish the dominance relationships. It is worth mentioning that their work deals with the CVRP version that does not fix the number of vehicles in advance.

## 3 Fundamentals

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

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

The CmVRP One-Commodity-Flow formulation (CF) was introduced alongside the problem itself by Kara et al. [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.

$$\text{(AL)} \quad \min \quad \sum_{a \in A} \sum_{\ell \in L} \tilde{d}_a \left(q^0 + \ell\right) y_a^\ell \tag{1a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^-(0)} \sum_{\ell \in L} y_a^\ell = K, \tag{1b}$$

$$\sum_{a \in \delta^+(0)} \sum_{\ell \in L} y_a^\ell = K, \tag{1c}$$

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

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

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

$$y_a^\ell = 0, \qquad \forall a \in \delta^+(0), \ell \in L \backslash 0, \tag{1g}$$

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

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

$$y_a^\ell \in \mathbb{Z}, \qquad \forall a \in A, \ell \in L. \tag{1j}$$

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

Constraints (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) enforce 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 $\mathcal{P}_{\text{AL}}$ denotes the LP relaxation of AL. Similarly, we define $\hat{z}^*$ subscribed by the name of the formulation to be the value of an optimal solution for the respective relaxed model.

**Theorem 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, $\hat{z}^*_{AL} \ge \hat{z}^*_{CF}$. 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 $v_i \neq v_{i+1}, \forall i = 1, \ldots, l-1$. We still enforce that the total demand of a route is at most $Q$ and, in the case of a $q$-route, the demand of each customer is accumulated as many times as it is visited. The set of all $q$-routes is denoted as $\Omega$.

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

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

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

$$(q\text{R}) \quad \min \quad \sum_{r \in \Omega} c_r \lambda_r \tag{2a}$$

$$\text{s.t.} \quad \sum_{r \in \Omega} \lambda_r = K, \tag{2b}$$

$$\sum_{r \in \Omega} h_{wr} \lambda_r = 1, \quad \forall w \in N, \tag{2c}$$

$$0 \leq \lambda_r \leq 1, \quad \forall r \in \Omega, \tag{2d}$$

$$\lambda_r \in \mathbb{Z}, \quad \forall r \in \Omega. \tag{2e}$$

The $h_{wr}$ coefficients represent the number of times a customer $w \in N$ is visited by a $q$-route $r \in \Omega$. The cost of a $q$-route $r$ is given by $c_r$ and the number of $q$-routes that can be traversed is constrained by (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 $q$R 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 $q$R is precisely justified by this method: the pricing subproblem of $q$-routes is weakly NP-hard, while pricing routes is strongly NP-hard. We also note that much stronger set partitioning formulations exist where additional constraints are imposed in $q$-routes. However, for the sake of simplicity, we restrict ourselves to only set partitioning formulations using $q$-routes, but our results can be easily adapted to these other stronger formulations.

Next, we present Theorem 2, that tightly links the AL and the $q$R formulations. Note that if a formulation F dominates a formulation G and vice-versa, we say that F and G are equivalent.

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

### *3.3 The Arc-Item Formulation*

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

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

The formulation is:

$$(\text{AI}) \quad \min \sum_{a \in A} \sum_{i \in V} \tilde{d}_a q^i x_a^i \tag{3a}$$

$$\text{s.t.} \quad \sum_{a \in \delta^+(0)} x_a^0 = K, \tag{3b}$$

$$\sum_{a \in \delta^-(w)} x_a^0 = 1, \qquad \forall w \in N, \tag{3c}$$

$$\sum_{a \in \delta^+(w)} x_a^0 = 1, \qquad \forall w \in N, \tag{3d}$$

$$\sum_{a \in \delta^-(t)} x_a^t = 0, \qquad \forall t \in N, \tag{3e}$$

$$\sum_{a\in\delta^+(t)} x_a^t = 1, \qquad \forall t \in N, \tag{3f}$$

$$\sum_{a\in\delta^-(w)} x_a^i = \sum_{a\in\delta^+(w)} x_a^i, \qquad \forall w \in N,\ i \in N\backslash w, \tag{3g}$$

$$x_a^i \le x_a^0, \qquad \forall a \in A,\ i \in N, \tag{3h}$$

$$\sum_{i\in N} q^i x_a^i \le Q, \qquad \forall a \in \delta^-(0), \tag{3i}$$

$$0 \le x_a^i \le 1, \qquad \forall a \in A,\ i \in V, \tag{3j}$$

$$x_a^i \in \mathbb{Z}, \qquad \forall a \in A,\ i \in V. \tag{3k}$$

We note that item 0 is always visited before any arcs in all routes and, therefore, variables $x_a^0$ just merely indicate if an arc $a$ is present in some route or not. The objective function (3a) minimizes the total cost of all the selected arc-items and, by considering $q^0$ as the curb weight, the objective function correctly considers the total weight (load plus curb weight) picked up prior to traversing $a$.

The selected arc-items whose items are indexed by 0 represent a full route in the classical sense, which also plays the role of a guide route: if an arc-item with item other than 0 is selected in an arc, then the arc-item with item 0 must also be selected in the same arc. With this in mind, we have that Constraint (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.*

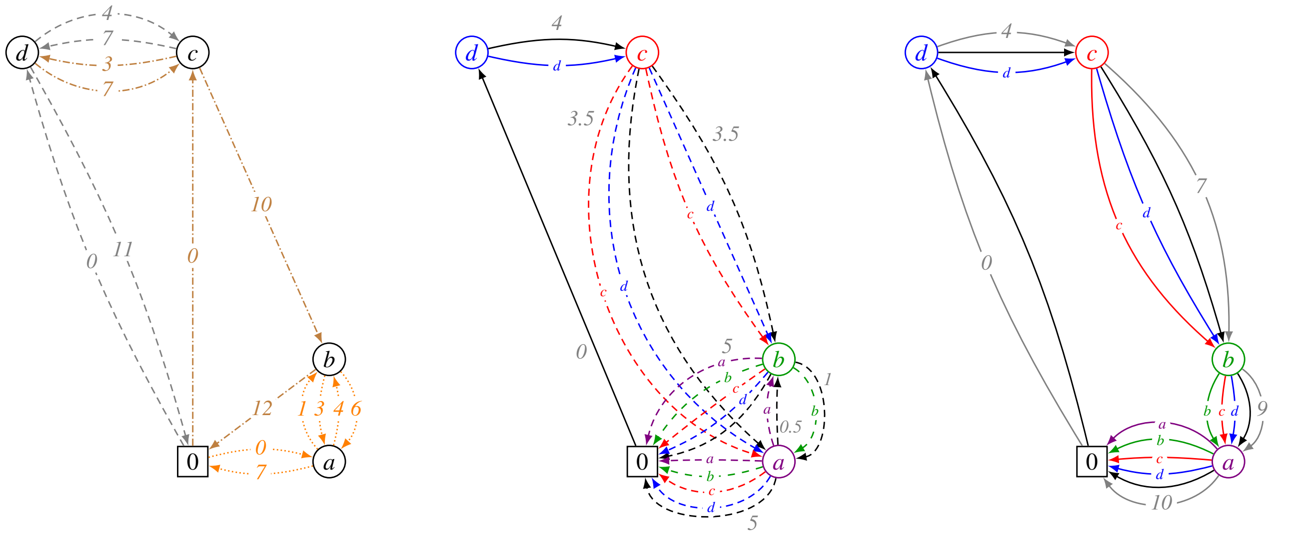
As a final note, we remark that the CmVRP AI formulation is very similar to other formulations to the CVRP found in the literature using variables that have the same purpose, e.g., the MCF1a (Letchford & Salazar-González [42]) and the MCF2b (Letchford & Salazar-González [42, 43]). Beyond the objective 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 MCF2b 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 $\sum_{\ell\in L} \ell y_a^\ell$ and in the AI formulation is $\sum_{i\in N} q^i x_a^i$, the formulation makes sure these quantities are the same. In Figure 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 $q$R 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.



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

It is not hard to extend the above analysis of Figure 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:

$$x_a^0 = \sum_{\ell \in L} y_a^\ell, \quad \forall a \in A, \tag{4a}$$

$$\sum_{i \in N} q^i x_a^i = \sum_{\ell \in L} \ell y_a^\ell, \quad \forall a \in A. \tag{4b}$$

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 $[x_a^i]$ and $[y_a^\ell]$, 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 $(x, y) \in \mathcal{P}_{AIL}$, there exist points $x \in \mathcal{P}_{AI}$ and $y \in \mathcal{P}_{AL}$ with the same objective value; particularly, $\hat{z}^*_{AIL} \geq \hat{z}^*_{AI}$ and $\hat{z}^*_{AIL} \geq \hat{z}^*_{AL}$. Moreover, there are instances where these dominance relations are strict.*

*Proof.* By construction, AIL dominates AI.

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

$$\sum_{a \in A} \sum_{i \in V} \tilde{d}_a q^i x_a^i = \sum_{a \in A} \tilde{d}_a \left( q^0 x_a^0 + \sum_{i \in N} q^i x_a^i \right) = \sum_{a \in A} \tilde{d}_a \left( q^0 \sum_{\ell \in L} y_a^\ell + \sum_{\ell \in L} \ell y_a^\ell \right) = \sum_{a \in A} \sum_{\ell \in L} \tilde{d}_a \left( q^0 + \ell \right) y_a^\ell,$$

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 $q$R formulation is equivalent to the AL. Therefore, we attempted to combine the $q$R and AI formulations, and this is what this subsection is devoted to.

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

$$x_a^i = \sum_{r \in \Omega} \dot{h}_{ar}^i \lambda_r, \quad \forall a \in A, i \in V, \tag{5}$$

where each one of the coefficients given by $\dot{h}_{ar}^i$ represents the number of times an item $i \in V$, picked up at vertex $i$, is carried over an arc $a \in A$ in a $q$-route $r \in \Omega$. In these constraints, we can see that the $[\dot{h}_{ar}^i]$ coefficients, summed through the selected $q$-routes/columns among all the $r \in \Omega$, were precisely tailored to match the $[x_a^i]$ 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 AI$q$R 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 AIqR dominates qR, that is, for any point $x \in \mathcal{P}_{AIqR}$, there exists a point $\lambda \in \mathcal{P}_{qR}$ with the same objective value; particularly, $\hat{z}^*_{AIqR} \geq \hat{z}^*_{qR}$. Moreover, there are instances where this dominance is strict.*

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

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

matching the objective function (2a) of the $q$R. Thus, we have that $z^*_{\mathrm{AI}q\mathrm{R}} \geq z^*_{q\mathrm{R}}$. Our computational experiments show that there are instances where $\hat{z}^*_{\mathrm{AI}q\mathrm{R}} > \hat{z}^*_{q\mathrm{R}}$, therefore, the main result follows. □

**Theorem 6** *The formulation AIqR dominates AI, that is, for any point $x \in \mathcal{P}_{AIqR}$, there exists a point $x \in \mathcal{P}_{AI}$ with the same objective value; particularly, $\hat{z}^*_{AIqR} \geq \hat{z}^*_{AI}$. Moreover, there are instances where this dominance is strict.*

*Proof.* The dominance of the AI$q$R over the AI is guaranteed by the former construction. By Theorem 3, the AI does not dominate $q$R, which finishes the proof. □

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

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

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

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

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

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

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

**Theorem 7** *The Prc-AIqR is strongly NP-hard.*

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

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

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

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

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

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

We note that the hardness proof relies on the fact that each arc-item combination has a different cost, which in turn comes from the fact that we have one coupling constraint (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 AIL$q$R, we start from the entire simplified AIL formulation (3a-3k, 1e-1j, 4a-4b) and the $q$R constraints (2b-2e). We couple these together by making use of the following relationship between the arc-loads ($[y_a^\ell]$) and the $q$-routes:

$$y_a^\ell = \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r, \qquad \forall a \in A, \ell \in L, \tag{6}$$

where the $\tilde{h}_{ar}^\ell$ coefficient represents the number of times a load $\ell \in L$ is carried over an arc $a \in A$ in a $q$-route $r \in \Omega$. Theorem 2 states that the AL and the $q$R formulations are equivalent, and thus we remove the AL constraints (1e-1h).

We now prove that constraints (2b-2c) from $q$R 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:

$$\sum_{a \in \delta^+(0)} x_a^0 = \sum_{a \in \delta^+(0)} \sum_{\ell \in L} y_a^\ell = \sum_{a \in \delta^+(0)} \sum_{\ell \in L} \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r = \sum_{r \in \Omega} \sum_{a \in \delta^+(0)} \sum_{\ell \in L} \tilde{h}_{ar}^\ell \lambda_r = \sum_{r \in \Omega} \lambda_r = K.$$

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

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

$$\sum_{a \in \delta^+(w)} x_a^0 = \sum_{a \in \delta^+(w)} \sum_{\ell \in L} y_a^\ell = \sum_{a \in \delta^+(w)} \sum_{\ell \in L} \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r = \sum_{r \in \Omega} \sum_{a \in \delta^+(w)} \sum_{\ell \in L} \tilde{h}_{ar}^\ell \lambda_r = \sum_{r \in \Omega} h_{wr} \lambda_r = 1.$$

This holds since $\sum_{a \in \delta^+(w)} \sum_{\ell \in L} \tilde{h}_{ar}^\ell$ 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:

$$x_a^0 = \sum_{\ell \in L} \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r, \qquad \forall a \in A, \tag{7a}$$

$$\sum_{i \in N} q^i x_a^i = \sum_{\ell \in L} \ell \sum_{r \in \Omega} \tilde{h}_{ar}^\ell \lambda_r, \qquad \forall a \in A. \tag{7b}$$

Finally, we remove the $[y_a^\ell]$ variables, as well as any other constraints (e.g. nonnegativity) involving them. Note that the model still implicitly uses the concept of arc-load to make the connection between the arc-items and the $q$-routes. Summing up, the AIL$q$R variables are the $[x_a^i]$ 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 AIL$q$R formulation implies the following theorem.

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

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

$$\bar{c}_r = c_r - \left( \sum_{a \in A} \left( - \sum_{\ell \in L} \tilde{h}_{ar}^\ell \right) \pi_a + \sum_{a \in A} \left( - \sum_{\ell \in L} \ell \tilde{h}_{ar}^r \right) \mu_a \right) = c_r + \sum_{a \in A} \sum_{\ell \in L} (\pi_a + \mu_a \ell) \tilde{h}_{ar}^\ell = \sum_{a \in A} \sum_{\ell \in L} (\pi_a + \mu_a \ell) \tilde{h}_{ar}^\ell.$$

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

$$(t\text{R}) \quad \min \sum_{t \in V} \sum_{r \in \Psi_t} c_r \lambda_r \tag{8a}$$

$$\text{s.t.} \quad \sum_{r \in \Psi_0} \lambda_r = K, \tag{8b}$$

$$\sum_{r \in \Psi_t} \lambda_r = 1, \qquad \forall t \in N, \tag{8c}$$

$$\sum_{r \in \Psi_t} \ddot{h}_{ar} \lambda_r \leq \sum_{s \in \Psi_0} \ddot{h}_{as} \lambda_s, \quad \forall a \in A,\ t \in N, \tag{8d}$$

$$0 \leq \lambda_r \leq 1, \qquad \forall r \in \Omega, \tag{8e}$$

$$\lambda_r \in \mathbb{Z}, \qquad \forall r \in \Omega. \tag{8f}$$

Comparing to the $q$R 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 $t$R formulation (AI$t$R). It comprises the entire AI model (3a-3k), all the constraints of the $t$R (8b-8f), and the coupling constraints

$$x_a^t = \sum_{r \in \Psi_t} \ddot{h}_{ar} \lambda_r, \quad \forall a \in A, t \in V. \tag{9}$$

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, $\hat{z}^*_{AItR} \geq \hat{z}^*_{AI}$. Moreover, there are instances where this dominance is strict.*

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

We can draw the following theorems from our experimental results.

**Theorem 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 AI$t$R. Note that in each column generation round, $n$ pricing subproblems will be run (one for each item). Let $\eta_a^t$, for all $a \in A$, $t \in V$, be the dual variables associated to the coupling constraints (9).

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

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

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

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

$n$ times, i.e., one for each item $t \in V$. This version of the problem is called Prc-AI$t$R, and it is weakly NP-hard, as it can be solved by $n$ calls to SPPRC.

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

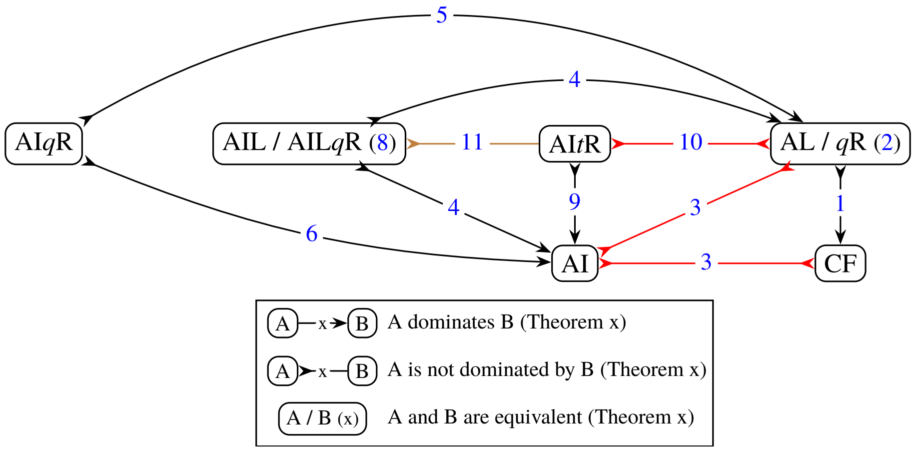



Figure 2. Some dominance relations among CmVRP formulations

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

## 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[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, AI and AIL formulations. Columns "Inst" and "$Q$" show, respectively, the name of the instance and the vehicle capacity $Q$. Column "#" presents the total number of instances. For each of the above formulations, the LP relaxation was solved and the results are presented in columns "$\hat{z}$", "T(s)" and "G(%)", which show, respectively, the optimal value, the time to solve it, and the gap. Note that the gap is calculated as follows: given the best known integer solution value ($z$) and a lower bound value $\hat{z}$, the gap G(%) is given by $((z - \hat{z})/z)100$. In fact, we make use of optimal integer solutions for the entire set of small instances, that were computed using the BCP program of Fukasawa et al. [25]. T(s) has a value of 0 when computations took less than 0.05 seconds. Finally, column "#I" presents a mark $*$ whenever the solution to the corresponding LP relaxation is integral, and the total number of those in the last row.

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

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

The results for the instances S-n05-k1 and S-n08-k2 are enough to imply that AI does not dominate nor is dominated by AL, thus proving part of Theorem 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$q$R. 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 $\hat{z}$ and gap values between AIL$q$R and AIL, as expected due to Theorem 8. We highlight the extremely high time to solve the LP relaxation of AIL$q$R for instance S-n25-k5. It takes a total of 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$q$R 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).

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

Table 2. Results of LP relaxation of AIL$q$R on small instances

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

It is worth pointing out that the LP relaxation of AIL$q$R is solved without enhancements such as cutting planes or column generation improvements such as dual stabilization [20, 49]. Our main goal is to show that the new formulations, AIL and AIL$q$R, provide both in theory and in practice better lower bounds than the previous ones existing in the literature, regardless of the processing time taken. Knowing that and the fact that their LP relaxations are pseudo-polynomial time solvable, one can implement such improvements to attempt to tighten the bound and decrease the computational cost.

We also performed a computational experiment with the other proposed column generation based formulation, namely AI$t$R. Table 3 presents those results on the small instances. When comparing to the average number of rounds of AIL$q$R (216.8), the same statistic in AI$t$R (185.2) is similar, albeit somewhat smaller. However, the number of generated columns is about seven times larger than in the former model, the CG time is about 40 times larger, and the LP time and the total time are about 60 times larger. The main justification of AI$t$R is that it has a pseudo-polynomial time pricing subproblem, such as AIL$q$R. However, even though the number of column generation rounds is almost always smaller, the consistently higher number of generated columns led to larger linear programs. Each linear program took an average of 214.8s for the S-n25-k5 instance, against the mean of 2.9s for the AIL$q$R. This gives us some evidence that decreasing the number of rounds may not be good if accompanied by a big increase in the number of generated columns.

Table 3. Results of LP relaxation of AI$t$R on small instances

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

These average numbers are mainly influenced by the test over the instance S-n25-k5. For the sake of completeness, we report some statistics about a test of S-n25-k5 limited to 86,400s, 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 AI$t$R for instance S-n04-k1 reaches the same value of $\hat{z} = 112.5$ as the other relaxed models tested, however the solution is fractional. From the results in Table 3, one can state that AI$t$R does not dominate the pair AIL$q$R and AIL (implying Theorem 11). These results also imply that AI$t$R does not dominate nor is dominated by AL (implying Theorem 10), as well as implying that AI does not dominate AI$t$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 AI$q$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 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 4. Results on gaps of the LP relaxations of AI, AL, and AIL

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

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

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

In Figure 3, we also present a plot of the cumulative frequency of gaps in the instances. A point $(x, y)$ in this figure represents that $y$ instances have a gap at most $x$%. One can see that the AIL relaxed formulation reaches a gap of at most 5% in 64 instances. For the other two formulations, AL and AI, only 50 and 11 instances have a gap of at most 5%, respectively. Indeed, this chart shows that the average data contained in Table 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

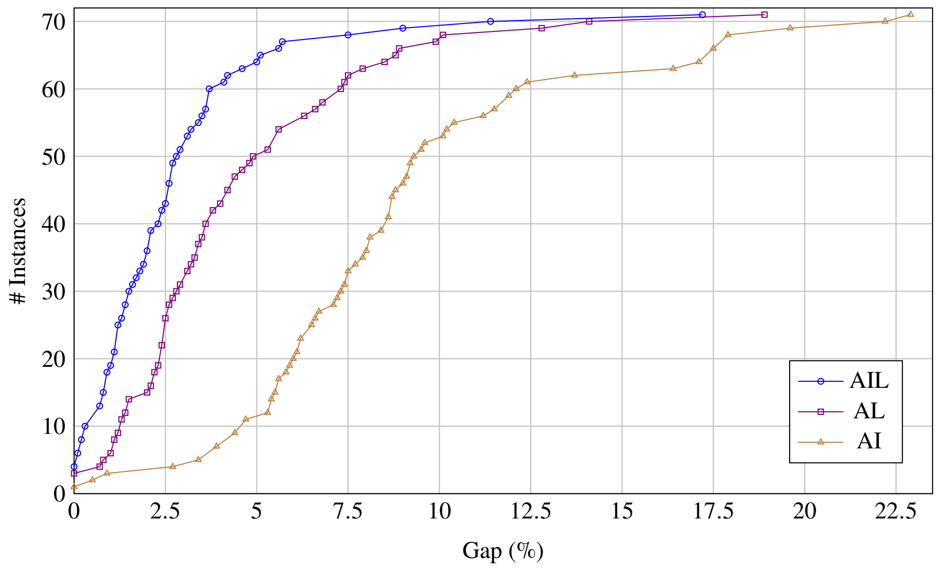



Figure 3. Gap cumulative frequency

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

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

## 6 Concluding Remarks

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

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

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

## Acknowledgments

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

## References

[1] H. Abeledo, R. Fukasawa, A. Pessoa, & E. Uchoa. The time dependent traveling salesman problem: Polyhedra and algorithm. *Mathematical Programming Computation*, 5(1):27–55, 2013. doi:10.1007/s12532-012-0047-y.

[2] F. Afrati, S. Cosmadakis, C. H. Papadimitriou, G. Papageorgiou, & N. Papakostantinou. The complexity of the travelling repairman problem. *RAIRO - Theoretical Informatics and Applications*, 20(1):79–87, 1986. doi:10.1051/ita/1986200100791.

[3] A. Archer, A. Levin, & D. P. Williamson. A faster, better approximation algorithm for the minimum latency problem. *SIAM Journal on Computing*, 37(5):1472–1498, 2008. doi:10.1137/07068151X.

[4] R. Baldacci, N. Christofides, & A. Mingozzi. An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts. *Mathematical Programming*, 115(2):351–385, 2008. doi:10.1007/s10107-007-0178-5.

[5] R. Baldacci, A. Mingozzi, & R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011. doi:10.1287/opre.1110.0975.

[6] M. L. Balinski & R. E. Quandt. On an integer program for a delivery problem. *Operations Research*, 12(2):300–304, 1964. doi:10.1287/opre.12.2.300.

[7] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, & P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):316–329, 1998. doi:10.1287/opre.46.3.316.

[8] L. Bianco, A. Mingozzi, & S. Ricciardelli. The traveling salesman problem with cumulative costs. *Networks*, 23(2): 81–91, 1993. doi:10.1002/net.3230230202.

[9] A. Blum, P. Chalasani, D. Coppersmith, B. Pulleyblank, P. Raghavan, & M. Sudan. The minimum latency problem. In *Proceedings of the 26$^{th}$ Annual ACM Symposium on Theory of Computing, STOC*, pages 163–171, New York, NY, US, 1994. ACM. doi:10.1145/195058.195125.

[10] T. Bulhões, R. Sadykov, & E. Uchoa. A branch-and-price algorithm for the minimum latency problem. *Computers & Operations Research*, 93:66–78, 2018. doi:10.1016/j.cor.2018.01.016.

[11] N. Christofides, A. Mingozzi, & P. Toth. Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations. *Mathematical Programming*, 20(1):255–282, 1981. doi:10.1007/BF01589353.

[12] D. Cinar, K. Gakis, & P. M. Pardalos. A 2-phase constructive algorithm for cumulative vehicle routing problems with limited duration. *Expert Systems with Applications*, 56:48–58, 2016. doi:10.1016/j.eswa.2016.02.046.

[13] D. Cinar, B. Cayir Ervural, K. Gakis, & P. M. Pardalos. Constructive algorithms for the cumulative vehicle routing problem with limited duration. In D. Cinar, K. Gakis, & P. M. Pardalos, editors, *Sustainable Logistics and Transportation: Optimization Models and Algorithms*, volume 129 of *SOIA*, pages 57–86. Springer, Cham, CH, 2017. doi:10.1007/978-3-319-69215-9_4.

[14] C. Contardo & R. Martinelli. A new exact algorithm for the multi-depot vehicle routing problem under capacity and route length constraints. *Discrete Optimization*, 12:129–146, 2014. doi:10.1016/j.disopt.2014.03.001.

[15] G. B. Dantzig & J. H. Ramser. The truck dispatching problem. *Management Science*, 6(1):80–91, 1959. doi:10.1287/mnsc.6.1.80.

[16] E. Demir, T. Bektaş, & G. Laporte. A review of recent research on green road freight transportation. *European Journal of Operational Research*, 237(3):775–793, 2014. doi:10.1016/j.ejor.2013.12.033.

[17] J. Desrosiers & M. E. Lübbecke. A primer in column generation. In G. Desaulniers, J. Desrosiers, & M. M. Solomon, editors, *Column Generation*, chapter 1, pages 1–32. Springer, Boston, MA, US, 2005. doi:10.1007/0-387-25486-2_1.

[18] K. Dorling, J. Heinrichs, G. G. Messier, & S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, 2017. doi:10.1109/TSMC.2016.2582745.

[19] M. Dror. Note on the complexity of the shortest path models for column generation in VRPTW. *Operations Research*, 42(5):977–978, 1994. doi:10.1287/opre.42.5.977.

[20] O. du Merle, D. Villeneuve, J. Desrosiers, & P. Hansen. Stabilized column generation. *Discrete Mathematics*, 194(1): 229–237, 1999. doi:10.1016/S0012-365X(98)00213-1.

[21] R. Eglese & T. Bektaş. Green vehicle routing. In P. Toth & D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 15, pages 437–458. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch15.

[22] J. Fakcharoenphol, C. Harrelson, & S. Rao. The k-traveling repairman problem. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 655–664, Philadelphia, PA, US, 2003. Society for Industrial and Applied Mathematics. doi:10.1145/1290672.1290677.

[23] J. Fakcharoenphol, C. Harrelson, & S. Rao. The k-traveling repairmen problem. *ACM Transactions on Algorithms*, 3 (4):40:1–40:16, 2007. doi:10.1145/1290672.1290677.

[24] R. Fukasawa, H. Longo, J. Lysgaard, M. Poggi de Aragão, M. Reis, E. Uchoa, & R. F. Werneck. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, 106(3):491–511, 2006. doi:10.1007/s10107-005-0644-x.

[25] R. Fukasawa, Q. He, & Y. Song. A branch-cut-and-price algorithm for the energy minimization vehicle routing problem. *Transportation Science*, 50(1):23–34, 2016. doi:10.1287/trsc.2015.0593.

[26] W. W. Garvin, H. W. Crandall, J. B. John, & R. A. Spellman. Applications of linear programming in the oil industry. *Management Science*, 3(4):407–430, 1957. doi:10.1287/mnsc.3.4.407.

[27] D. R. Gaur & R. R. Singh. A heuristic for cumulative vehicle routing using column generation. *Discrete Applied Mathematics*, 228:140–157, 2017. doi:10.1016/j.dam.2016.05.030.

[28] D. R. Gaur, A. Mudgal, & R. R. Singh. Routing vehicles to minimize fuel consumption. *Operations Research Letters*, 41(6):576–580, 2013. doi:10.1016/j.orl.2013.07.007.

[29] D. R. Gaur, A. Mudgal, & R. R. Singh. Approximation algorithms for cumulative VRP with stochastic demands. In *Proceedings of the 2nd International Conference on Algorithms and Discrete Applied Mathematics, CALDAM*, volume 9602 of *LNCS*, pages 176–189, Cham, CH, 2016. Springer. doi:10.1007/978-3-319-29221-2_15.

[30] D. R. Gaur, A. Mudgal, & R. R. Singh. Improved approximation algorithms for cumulative VRP with stochastic demands. *Discrete Applied Mathematics*, 2018. doi:10.1016/j.dam.2018.01.012.

[31] B. Golden, S. Raghavan, & E. Wasil, editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*, volume 43 of *ORCS*. Springer, Boston, MA, US, 2008. doi:10.1007/978-0-387-77778-8.

[32] M. Hassanalian & A. Abdelkefi. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 2017. doi:10.1016/j.paerosci.2017.04.003.

[33] H. Hernández-Pérez & J.-J. Salazar-González. The multi-commodity one-to-one pickup-and-delivery traveling salesman problem. *European Journal of Operational Research*, 196(3):987–995, 2009. doi:10.1016/j.ejor.2008.05.009.

[34] S. Irnich & G. Desaulniers. Shortest path problems with resource constraints. In G. Desaulniers, J. Desrosiers, & M. M. Solomon, editors, *Column Generation*, chapter 2, pages 33–65. Springer, Boston, MA, US, 2005. doi:10.1007/0-387-25486-2_2.

[35] S. Irnich, P. Toth, & D. Vigo. The family of vehicle routing problems. In P. Toth & D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 1, pages 1–33. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch1.

[36] R. Jothi & B. Raghavachari. Approximating the k-traveling repairman problem with repairtimes. *Journal of Discrete Algorithms*, 5(2):293–303, 2007. doi:10.1016/j.jda.2006.03.023.

[37] İ. Kara, B. Y. Kara, & M. K. Yetiş. Energy minimizing vehicle routing problem. In *Proceedings of the 1st International Conference on Combinatorial Optimization and Applications, COCOA*, volume 4616 of *LNCS*, pages 62–71, Berlin/Heidelberg, DE, 2007. Springer. doi:10.1007/978-3-540-73556-4_9.

[38] İ. Kara, B. Y. Kara, & M. K. Yetiş. Cumulative vehicle routing problems. In T. Caric & H. Gold, editors, *Vehicle Routing Problem*, chapter 6. IntechOpen, Rijeka, HR, 2008. doi:10.5772/5812.

[39] R. Kramer, A. Subramanian, T. Vidal, & L. d. A. F. Cabral. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, 243(2):523–539, 2015. doi:10.1016/j.ejor.2014.12.009.

[40] A. Langevin, F. Soumis, & J. Desrosiers. Classification of travelling salesman problem formulations. *Operations Research Letters*, 9(2):127–132, 1990. doi:10.1016/0167-6377(90)90052-7.

[41] A. N. Letchford & J.-J. Salazar-González. Projection results for vehicle routing. *Mathematical Programming*, 105 (2-3):251–274, 2006. doi:10.1007/s10107-005-0652-x.

[42] A. N. Letchford & J.-J. Salazar-González. Stronger multi-commodity flow formulations of the capacitated vehicle routing problem. *European Journal of Operational Research*, 244(3):730–738, 2015. doi:10.1016/j.ejor.2015.02.028.

[43] A. N. Letchford & J.-J. Salazar-González. The capacitated vehicle routing problem: Stronger bounds in pseudo-polynomial time. *European Journal of Operational Research*, 272(1):24–31, 2019. doi:10.1016/j.ejor.2018.06.002.

[44] A. Lucena. Time-dependent traveling salesman problem–the deliveryman case. *Networks*, 20(6):753–763, 1990. doi:10.1002/net.3230200605.

[45] I. Méndez-Díaz, P. Zabala, & A. Lucena. A new formulation for the traveling deliveryman problem. *Discrete Applied Mathematics*, 156(17):3223–3237, 2008. doi:10.1016/j.dam.2008.05.009.

[46] M. H. Mulati & F. K. Miyazawa. Tighter analysis of an approximation for the cumulative VRP. In *Anais do II Encontro de Teoria da Computação – XXXVII Congresso da Sociedade Brasileira de Computação, ETC*, pages 150–153, Porto Alegre, RS, BR, 2017. SBC. doi:10.5753/etc.2017.3202.

[47] A. Otto, N. Agatz, J. Campbell, B. Golden, & E. Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018. doi:10.1002/net.21818.

[48] D. Pecin, A. Pessoa, M. Poggi, & E. Uchoa. Improved branch-cut-and-price for capacitated vehicle routing. *Mathematical Programming Computation*, 9(1):61–100, 2017. doi:10.1007/s12532-016-0108-8.

[49] A. Pessoa, R. Sadykov, E. Uchoa, & F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, 30(2):339–360, 2018. doi:10.1287/ijoc.2017.0784.

[50] A. Pessoa, R. Sadykov, E. Uchoa, & F. Vanderbeck. A generic exact solver for vehicle routing and related problems. *Mathematical Programming*, 2020. doi:10.1007/s10107-020-01523-z.

[51] J.-C. Picard & M. Queyranne. The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling. *Operations Research*, 26(1):86–110, 1978. doi:10.1287/opre.26.1.86.

[52] M. Poggi & E. Uchoa. New exact algorithms for the capacitated vehicle routing problem. In P. Toth & D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 3, pages 59–86. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch3.

[53] M. Poggi de Aragão & E. Uchoa. Integer program reformulation for robust branch-and-cut-and-price algorithms. In *Proceedings of the Conference Mathematical Program in Rio: A Conference in Honour of Nelson Maculan*, pages 56–61, 2003. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.1278`.

[54] L. D. P. Pugliese & F. Guerriero. A survey of resource constrained shortest path problems: Exact solution approaches. *Networks*, 62(3):183–200, 2013. doi:10.1002/net.21511.

[55] F. Semet, P. Toth, & D. Vigo. Classical exact algorithms for the capacitated vehicle routing problem. In P. Toth & D. Vigo, editors, *Vehicle Routing: Problems, Methods, and Applications*, MOS-SIAM Series on Optimization, chapter 2, pages 37–57. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.ch2.

[56] R. R. Singh & D. R. Gaur. Cumulative VRP: A simplified model of green vehicle routing. In D. Cinar, K. Gakis, & P. M. Pardalos, editors, *Sustainable Logistics and Transportation: Optimization Models and Algorithms*, volume 129 of *SOIA*, pages 39–55. Springer, Cham, CH, 2017. doi:10.1007/978-3-319-69215-9_3.

[57] P. Toth & D. Vigo, editors. *Vehicle Routing: Problems, Methods, and Applications*. MOS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics, Philadelphia, PA, US, 2nd edition, 2014. doi:10.1137/1.9781611973594.

[58] S. Wang, M. Liu, & F. Chu. Approximate and exact algorithms for an energy minimization traveling salesman problem. *Journal of Cleaner Production*, 249:119433, 2020. doi:10.1016/j.jclepro.2019.119433.

[59] Y. Xiao, Q. Zhao, I. Kaku, & Y. Xu. Development of a fuel consumption optimization model for the capacitated vehicle routing problem. *Computers & Operations Research*, 39(7):1419–1431, 2012. doi:10.1016/j.cor.2011.08.013.

[60] E. E. Zachariadis, C. D. Tarantilis, & C. T. Kiranoudis. The load-dependent vehicle routing problem and its pick-up and delivery extension. *Transportation Research Part B: Methodological*, 71:158–181, 2015. doi:10.1016/j.trb.2014.11.004.

## 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 6. AI, AL, and AIL relaxed models full results with regular instances

| | | | Arc-Item | | | | Arc-Load | | | | Arc-Item-Load | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Inst | $Q$ | # | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) | T(s) | $\hat{z}$ | #I | G(%) |
| A-n32-k5 | 100 | | 1.1 | 39,989.20 | | 2.7 | 4.4 | 39,696.83 | | 3.4 | 709.9 | 40,706.20 | | 0.9 |
| A-n33-k5 | 100 | | 1.0 | 31,501.52 | | 5.6 | 4.1 | 32,539.13 | | 2.5 | 1,459.8 | 32,851.38 | | 1.5 |
| A-n33-k6 | 100 | | 1.2 | 34,983.37 | | 5.5 | 3.9 | 35,797.29 | | 3.3 | 3,486.5 | 36,739.79 | | 0.8 |
| A-n34-k5 | 100 | | 2.1 | 37,227.31 | | 9.5 | 8.1 | 38,553.10 | | 6.3 | 4,229.3 | 39,967.98 | | 2.8 |
| A-n36-k5 | 100 | | 2.9 | 37,098.70 | | 5.6 | 19.5 | 38,517.66 | | 2 | 2,146.2 | 38,804.80 | | 1.2 |
| A-n37-k5 | 100 | | 2.1 | 29,641.05 | | 4.4 | 21.6 | 29,820.92 | | 3.8 | 3,032.1 | 30,716.29 | | 0.9 |
| A-n37-k6 | 100 | | 4.1 | 45,749.12 | | 8.7 | 11.9 | 48,001.58 | | 4.2 | 1,427.9 | 48,559.70 | | 3.1 |
| A-n38-k5 | 100 | | 2.3 | 33,965.74 | | 11.9 | 13.2 | 34,656.54 | | 10.1 | 12,753.8 | 36,363.12 | | 5.7 |
| A-n39-k5 | 100 | | 8.5 | 39,528.51 | | 7.5 | 27.4 | 41,663.94 | | 2.5 | 4,124.5 | 42,058.45 | | 1.6 |
| A-n39-k6 | 100 | | 1.9 | 38,660.46 | | 6.5 | 5.7 | 40,253.84 | | 2.7 | 2,420.7 | 41,069.33 | | 0.7 |
| A-n44-k6 | 100 | | 10.2 | 45,311.43 | | 5.4 | 15.8 | 47,329.91 | | 1.1 | 7,345.8 | 47,755.22 | | 0.3 |

*Table 6 – continuation*

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A-n45-k6 | 100 | | 7.5 | 45,499.19 | | 12.4 | 15.7 | 48,706.45 | | 6.3 | 9,757.0 | 49,802.76 | | 4.1 |
| A-n45-k7 | 100 | | 15.2 | 58,971.15 | | 5.3 | 11.9 | 60,871.63 | | 2.2 | 11,519.9 | 61,363.06 | | 1.4 |
| A-n46-k7 | 100 | | 11.6 | 45,784.79 | | 4.7 | 18.6 | 47,458.33 | | 1.2 | 9,597.5 | 47,927.25 | | 0.2 |
| A-n48-k7 | 100 | | 23.6 | 54,841.82 | | 6.2 | 21.8 | 57,077.72 | | 2.4 | 17,725.9 | 57,315.46 | | 2 |
| A-n53-k7 | 100 | | 33.6 | 50,902.54 | | 8.8 | 38.0 | 53,035.17 | | 4.9 | 34,624.5 | 54,194.67 | | 2.9 |
| A-n54-k7 | 100 | | 56.1 | 59,115.59 | | 7.2 | 31.8 | 62,061.14 | | 2.6 | 65,921.4 | 62,936.50 | | 1.2 |
| A-n55-k9 | 100 | | 13.0 | 54,660.22 | | 6.2 | 19.1 | 55,825.30 | | 4.2 | 32,250.2 | 56,877.71 | | 2.4 |
| A-n60-k9 | 100 | | 83.1 | 68,282.56 | | 8.1 | 39.4 | 72,488.48 | | 2.5 | 81,763.9 | 72,838.46 | | 2 |
| A | | 19 | 14.8 | | | 7 | 17.5 | | | 3.6 | 16,120.9 | | | 1.9 |
| B-n31-k5 | 100 | | 1.3 | 33,827.11 | | 3.9 | 3.4 | 33,331.46 | | 5.3 | 2,519.7 | 34,590.88 | | 1.8 |
| B-n34-k5 | 100 | | 1.5 | 40,083.49 | | 9.2 | 6.5 | 41,236.60 | | 6.6 | 7,022.0 | 42,959.26 | | 2.7 |
| B-n35-k5 | 100 | | 1.1 | 47,969.38 | | 9.3 | 8.9 | 48,167.65 | | 8.9 | 885.9 | 48,947.42 | | 7.5 |
| B-n38-k6 | 100 | | 2.6 | 39,353.87 | | 4.4 | 5.0 | 39,854.33 | | 3.2 | 3,765.7 | 40,216.89 | | 2.3 |
| B-n39-k5 | 100 | | 1.6 | 27,112.00 | | 7.4 | 5.4 | 27,085.61 | | 7.5 | 11,870.4 | 28,178.21 | | 3.7 |
| B-n41-k6 | 100 | | 2.4 | 41,653.42 | | 10.1 | 7.0 | 44,088.08 | | 4.8 | 4,617.0 | 45,211.06 | | 2.4 |
| B-n43-k6 | 100 | | 11.4 | 36,572.45 | | 8.6 | 12.3 | 38,261.48 | | 4.4 | 11,224.6 | 39,022.54 | | 2.5 |
| B-n44-k7 | 100 | | 3.7 | 47,587.28 | | 8.4 | 7.9 | 49,995.45 | | 3.8 | 21,155.5 | 50,535.65 | | 2.7 |
| B-n45-k5 | 100 | | 5.2 | 33,457.47 | | 10.2 | 12.5 | 34,316.92 | | 7.9 | 23,516.7 | 35,164.71 | | 5.6 |
| B-n45-k6 | 100 | | 9.8 | 34,623.87 | | 11.2 | 28.0 | 35,143.17 | | 9.9 | 35,988.5 | 37,661.95 | | 3.4 |
| B-n50-k7 | 100 | | 3.1 | 36,603.71 | | 7.7 | 25.8 | 37,440.41 | | 5.6 | 68,479.1 | 38,008.92 | | 4.2 |
| B-n50-k8 | 100 | | 58.2 | 65,631.61 | | 8.6 | 35.2 | 70,858.26 | | 1.3 | 67,816.9 | 70,980.11 | | 1.1 |
| B-n51-k7 | 100 | | 8.4 | 52,660.93 | | 22.2 | 34.3 | 54,949.21 | | 18.9 | 66,034.3 | 56,097.61 | | 17.2 |
| B-n52-k7 | 100 | | 13.2 | 36,592.44 | | 8.7 | 51.9 | 37,140.49 | | 7.3 | 59,713.7 | 38,575.42 | | 3.7 |
| B-n67-k10 | 100 | | 121.8 | 53,073.96 | | 6.7 | 36.8 | 54,429.07 | | 4.4 | 77,429.0 | 54,943.50 | | 3.5 |
| B | | 15 | 16.4 | | | 9.1 | 18.7 | | | 6.7 | 30,802.6 | | | 4.3 |
| E-n13-k4 | 6000 | | 0.0 | 728,517.91 | | 6.5 | 0.0 | 774,300.00 | | 0.7 | 0.4 | 778,066.67 | | 0.2 |
| E-n22-k4 | 6000 | | 0.2 | 1,131,170.00 | | 5.4 | 0.2 | 1,178,581.54 | | 1.4 | 59.6 | 1,195,200.00 | * | 0 |
| E-n30-k3 | 4500 | | 1.0 | 1,003,393.75 | | 19.6 | 15.5 | 1,071,232.83 | | 14.1 | 9,316.0 | 1,105,568.11 | | 11.4 |
| E-n31-k7 | 140 | | 4.1 | 23,255.65 | | 9.1 | 5.7 | 25,327.38 | | 1 | 800.0 | 25,516.77 | | 0.3 |
| E | | 4 | 1.3 | | | 10.2 | 5.4 | | | 4.3 | 2,544.0 | | 1 | 3 |
| P-n16-k8 | 35 | | 0.0 | 8,389.56 | | 0.5 | 0.0 | 8,434.50 | * | 0 | 0.1 | 8,434.50 | * | 0 |
| P-n19-k2 | 160 | | 0.1 | 15,644.75 | | 17.5 | 2.0 | 17,352.67 | | 8.5 | 325.1 | 18,022.95 | | 5 |
| P-n20-k2 | 160 | | 0.2 | 16,052.25 | | 17.9 | 3.3 | 18,137.00 | | 7.3 | 604.3 | 18,847.06 | | 3.7 |
| P-n21-k2 | 160 | | 0.3 | 15,757.56 | | 17.5 | 3.3 | 17,803.14 | | 6.8 | 673.0 | 18,523.52 | | 3.1 |
| P-n22-k2 | 160 | | 0.4 | 16,116.74 | | 17.9 | 4.3 | 18,541.17 | | 5.6 | 1,132.2 | 19,214.07 | | 2.1 |
| P-n22-k8 | 3000 | | 0.1 | 914,620.43 | | 7.9 | 0.0 | 978,408.33 | | 1.5 | 2.1 | 985,706.28 | | 0.7 |
| P-n23-k8 | 40 | | 0.1 | 11,016.91 | | 8.7 | 0.0 | 11,818.40 | | 2.1 | 3.0 | 11,818.40 | | 2.1 |
| P-n40-k5 | 140 | | 8.8 | 28,967.73 | | 11.9 | 18.0 | 31,872.52 | | 3.1 | 12,347.6 | 32,321.42 | | 1.7 |
| P-n45-k5 | 150 | | 15.4 | 34,035.61 | | 13.7 | 52.9 | 38,121.14 | | 3.4 | 28,895.4 | 38,874.85 | | 1.5 |
| P-n50-k7 | 150 | | 15.6 | 39,058.48 | | 7.3 | 81.4 | 41,640.86 | | 1.1 | 40,342.6 | 42,070.63 | | 0.1 |
| P-n50-k8 | 120 | | 12.9 | 36,205.92 | | 10.4 | 70.1 | 38,815.08 | | 4 | 15,501.5 | 38,965.98 | | 3.6 |
| P-n50-k10 | 100 | | 4.3 | 34,219.82 | | 5.9 | 16.4 | 35,506.50 | | 2.4 | 7,054.7 | 35,682.50 | | 1.9 |
| P-n51-k10 | 80 | | 8.6 | 29,397.99 | | 6.6 | 17.8 | 30,581.66 | | 2.9 | 6,844.1 | 30,663.97 | | 2.6 |
| P-n55-k7 | 170 | | 25.1 | 43,962.25 | | 9.6 | 133.0 | 47,344.28 | | 2.6 | 81,615.1 | 48,199.70 | | 0.9 |
| P-n55-k8 | 160 | | 15.2 | 43,072.41 | | 7.1 | 143.0 | 45,304.51 | | 2.3 | 75,211.6 | 45,756.31 | | 1.4 |
| P-n55-k10 | 115 | | 14.6 | 38,583.67 | | 4.7 | 29.7 | 39,913.55 | | 1.5 | 11,349.7 | 40,164.92 | | 0.8 |
| P-n55-k15 | 70 | | 8.1 | 33,966.00 | | 9 | 3.0 | 36,382.37 | | 2.5 | 8,582.2 | 36,512.91 | | 2.1 |
| P-n60-k10 | 120 | | 21.8 | 43,579.44 | | 6 | 83.5 | 45,265.45 | | 2.4 | 56,595.7 | 45,814.96 | | 1.2 |
| P-n60-k15 | 80 | | 10.6 | 39,354.71 | | 3.9 | 8.8 | 40,412.26 | | 1.3 | 11,544.0 | 40,519.06 | | 1 |
| P-n65-k10 | 130 | | 68.9 | 49,740.25 | | 6.1 | 88.4 | 51,819.86 | | 2.2 | 85,160.6 | 52,353.50 | | 1.2 |
| P | | 20 | 11.6 | | | 9.5 | 37.9 | | 1 | 3.2 | 22,189.2 | | 1 | 1.8 |
| ulysses-n16-k3 | 5 | | 0.0 | 18,566.48 | | 8.1 | 0.0 | 19,482.88 | | 3.6 | 0.6 | 19,994.65 | | 1.1 |
| gr-n17-k3 | 6 | | 0.0 | 7,265.66 | | 0.9 | 0.0 | 7,331.20 | * | 0 | 0.4 | 7,331.20 | * | 0 |
| gr-n21-k3 | 7 | | 0.1 | 11,756.40 | * | 0 | 0.0 | 11,756.40 | * | 0 | 0.5 | 11,756.40 | * | 0 |
| ulysses-n22-k4 | 6 | | 0.2 | 22,919.50 | | 5.8 | 0.0 | 23,202.42 | | 4.6 | 1.6 | 23,682.88 | | 2.6 |
| gr-n24-k4 | 7 | | 0.7 | 6,899.19 | | 3.4 | 0.0 | 7,081.20 | | 0.8 | 7.0 | 7,093.50 | | 0.7 |
| fri-n26-k3 | 10 | | 0.8 | 6,377.11 | | 7.5 | 0.0 | 6,704.69 | | 2.8 | 15.6 | 6,887.02 | | 0.1 |
| bayg-n29-k4 | 8 | | 0.7 | 7,598.30 | | 12.1 | 0.0 | 8,334.38 | | 3.6 | 30.6 | 8,422.20 | | 2.6 |
| bays-n29-k5 | 6 | | 0.9 | 8,560.17 | | 8 | 0.0 | 8,975.16 | | 3.5 | 12.1 | 9,011.33 | | 3.2 |
| dantzig-n42-k4 | 11 | | 46.1 | 5,780.02 | | 9.2 | 0.1 | 6,165.44 | | 3.1 | 818.2 | 6,279.93 | | 1.3 |
| swiss-n42-k5 | 9 | | 7.2 | 6,935.52 | | 11.5 | 0.1 | 7,392.63 | | 5.6 | 526.4 | 7,622.63 | | 2.7 |
| gr-n48-k3 | 16 | | 117.7 | 40,772.70 | | 22.9 | 0.6 | 46,124.96 | | 12.8 | 3,899.9 | 48,122.04 | | 9 |
| att-n48-k4 | 15 | | 34.3 | 256,206.82 | | 17.1 | 0.3 | 281,740.34 | | 8.8 | 5,043.2 | 294,974.97 | | 4.6 |
| hk-n48-k4 | 15 | | 224.5 | 97,256.11 | | 16.4 | 0.4 | 107,752.58 | | 7.4 | 3,435.1 | 110,324.01 | | 5.1 |

*Table 6 – continuation*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| V | 13 | 33.3 | 1 | 9.5 | 0.1 | 2 | 4.4 | 1,060.9 | 2 | 2.5 |
| | 71 | 16.8 | 1 | 8.8 | 19.6 | 3 | 4.3 | 17,409.7 | 4 | 2.6 |