

Fleet Size and Mix Split-Delivery Vehicle Routing

A study of MIP formulations with CP integration

Arthur Mahéo · Tommaso Urli ·
Phil Kilby

Received: date / Accepted: date

Abstract In the classic Vehicle Routing Problem (VRP) a fleet of vehicles has to visit a set of customers while minimising the operations' costs. We study a rich variant of the VRP featuring split deliveries, an heterogeneous fleet, and vehicle-commodity incompatibility constraints. Our goal is twofold: define the cheapest routing and the most adequate fleet.

To do so, we split the problem into two interdependent components: a fleet design component and a routing component. First, we define two Mixed Integer Programming (MIP) formulations for each component. Then we discuss several improvements in the form of valid cuts and symmetry breaking constraints.

The main contribution of this paper is a comparison of the four resulting models for this Rich VRP. We highlight their strengths and weaknesses with extensive experiments.

Finally, we explore a lightweight integration with Constraint Programming (CP). We use a fast CP model which gives good solutions and use the solution to warm-start our models.

Keywords Rich Vehicle Routing, Split Delivery, Fleet Size and Mix, Mixed Integer Programming, Constraint Programming

Mathematics Subject Classification (2010) 90B06 · 90C11

1 Introduction

Beyond routing their vehicles at the least cost, goods carriers often want to know what is the best fleet to use. For example, some vehicle types are more expensive, in terms of maintenance and usage than others. Our work is based on a tender for grocery delivery in Queensland, Australia. A goods carrier, handling delivery of chilled and ambient temperature products, provided us with a year of demand

The Australian National University · Data61, Canberra City
Level 3 Tower A, 7 London Circuit, City ACT 2601
E-mail: {Arthur.Mahéo;Tommaso.Urli;Philip.Kilby}@data61.csiro.au

data. We want to determine what is the best fleet and the cheapest routing to satisfy customers.

Our main contribution is to compare different modelling approaches, and provide extensive computational experiments based on real-world instances. The ultimate goal of this project is to find good modelling techniques for a more general multi-day setting. In this work we focus on the single day setting as a first step.

The Vehicle Routing Problem (VRP, Dantzig and Ramser, 1959; Clarke and Wright, 1964) consists in routing a fleet of vehicles to visit a set of customers. It was first defined as an extension of the Travelling Salesman Problem (TSP) with multiple salespeople. As such, it belongs to the class of NP-hard problems (Lenstra and Kan, 1981). Its most common extension is the Capacitated VRP (CVRP, Toth and Vigo, 2001). In this case, customers have an associated demand and vehicles have a limited capacity.

Another common extension is the Split Delivery VRP (SDVRP, Dror and Trudeau, 1989). In the classic VRP, customers are visited exactly once. The SDVRP relaxes this constraint, allowing more than one visit per customer. It is especially useful when demands exceed the vehicle capacity. Although it is a relaxation, it has been proven to be harder because because of the additional degree of freedom in planning the deliveries. Despite its complexity, there are good reasons for studying the SDVRP as it can lead to significant savings.

In recent years, driven by industrial needs, several realistic variants of the VRP appeared. First, models were extended to handle multiple commodities (types of goods). Usually such extension comes with compatibility issues, as vehicles cannot transport all commodities. A more ambitious extension is the Fleet Size and Mix VRP (FSMVRP) that considers finding a good fleet size and composition.

We call Rich VRP a problem merging many such extensions, see Caceres-Cruz et al (2014) for a survey. In this paper we consider the fleet size and mix, multi-commodity, capacitated, and split delivery VRP.

Our current work tries to address issues raised in the experimental analysis of Kilby and Urli (2016). We propose a stronger MIP scheme to compare with their CP approach.

To solve this problem, we focus on the two distinct, but interdependent parts of the problem. First is the fleet design component, and second is the routing component. Bigger vehicles with better compatibility are more expensive to run. We propose two different MIP models for each component, amounting to four complete MIPs.

Finally, we propose a lightweight integration of a CP model into our solving process. The main strength of the CP model is to be able to find good solution fast. Thus, we use it to prime our MIP search. Instead of letting the solver find a starting solution, we use the CP model to find one.

2 Related Work

In this section we review some of the relevant literature on SDVRP, FSMVRP, and RVRP formulations, and we consider some works on valid inequalities, some of which we adapt to work with our models.

In Dror and Trudeau (1989) the authors present SDVRP together with an analysis of the problem's complexity and the potential savings that can be attained

by using it as a delivery scheme. The conclusion of this work is that, despite the SDVRP being a relaxation of the VRP, it is harder to solve. However, the additional complexity pays off, since allowing split deliveries can lead to substantial gains in terms of distances and number of vehicles needed, especially when the demand of the customers is slightly less than a full truckload. Our work extends the above, in that we consider additional compatibility constraints and a multi-commodity scenario.

Regarding RVRP formulations we refer the interested reader to a recent, excellent survey by Caceres-Cruz et al (2014).

Concerning exact algorithms to solve the VRP under capacity or time window constraints, the interested reader can refer to Baldacci et al (2012). The authors present a comparison of different exact formulations. Two main class of algorithms emerge: branch-and-cut algorithms and set covering algorithms. The former are based on the idea of adding valid cuts during a branch-and-bound search. The latter employ an exponential number of variables, and are therefore solved using column generation. Our work can be seen as a preliminary study to compare MIP models to be used in a decomposition scheme.

Regarding the research on valid inequalities for VRP, the literature is quite rich. In an extension (Dror et al, 1994) of Dror and Trudeau (1989), the authors focus on improving the solution process by devising valid cuts for their previous model. Both works are based on the traditional CVRP, where a single commodity is considered, and the fleet is fixed.

In Letchford and Salazar-González (2006), a study on the relative strengths of different classes of inequalities is presented, this time for the CVRP. In a follow-up to this work (Letchford and Salazar-González, 2015) the authors add two new multi-commodity formulations, and study additional families of inequalities. In Baldacci et al (2012) a number of valid cuts are also discussed; often these are not trivial as few of them have polynomial algorithms available. Our approach adopts several of the valid inequalities presented in the above works.

The FSMVRP is a variant of the Heterogeneous VRP (HVRP) where the number of vehicles is not fixed. The HVRP has been first defined in the work of Golden et al (1984) as a relaxation of the CVRP where the vehicles do not have a fixed type. The authors discuss a number of heuristic algorithms, some based on the Clarke and Wright Savings algorithm, some based on the concept of *opportunity cost*, a way to evaluate the potential gain by using a vehicle of a different type.

Most of the works on FSMVRP are limited to single-commodity and single-day scenarios, furthermore very few exact algorithms exist in the literature, the main focus being on improving the quality of linear relaxations in order to strengthen bounding procedures – e.g. Jabali et al (2012).

Vidal et al (2014) use a component-based framework to solve a variety real-world VRP variants, including fleet size and mix vehicle routing, is proposed. The approach, however, does not handle split deliveries or multiple commodities.

An exact algorithm was proposed by Baldacci and Mingozzi (2009) for the HVRP. Their approach is based on a set partitioning formulation, which they solve using a reduced model in combination with a bounding procedure. Although this approach encompasses a wide variety of possible variations of the HVRP, it does not handle split deliveries or multiple commodities.

Yaman (2006) offers a work similar to ours in the sense that they design and compare six different formulations for the HVRP: four based on the Miller-Tucker-Zemlin constraints and two based on a flow formulation. They design a number of valid inequalities for the different models, then report computational experiments using standard benchmarks from the literature. However, they only provide the time and value of the linear relaxations, no exact result is provided.

For a recent review on different FSMVRP algorithms and problem classification, we refer the interested reader to a survey by Koç et al (2016). Following the classification proposed in another review by Baldacci et al (2008) our problem is a FSMFD, with unlimited fleet size, fixed costs, and routing costs.

To the best of our knowledge the work of Kilby and Urli (2016) is the only combining FSMVRP with split deliveries and multiple commodities. The authors present a CP and Large Neighbourhood Search (LNS) approach to solve the same problem we address here. The experimental analysis revealed the strength of MIP on some particular problem configurations, and therefore motivated the present research. Moreover, we employ the CP model presented in that work to prime our MIP formulation, so as to increase the performance of our approach.

3 Problem

In this section we provide a description of the problem components: entities, constraints, and objective. We defer the description of routing and fleet constraints to ?? as they are specific to each model. We decided to limit the use of equations in this section because they are often very similar. ?? presents the complete mathematical formulation of the resulting models which combine: the objective function, the core constraints, one fleet model, and one routing model. ?? presents an example augmented with the constraints.

3.1 Entities

dep	the depot where the vehicles start and end their routes;
K	the set of product types (or <i>commodities</i>) that must be dispatched;
T	the set of vehicle types that can be used to deliver the demand, which differ in capacity and cost;
cap_t	the total capacity of a type of vehicle (across all the commodities);
C	the set of customers to be served. We also denote by $\mathbf{C}' = \mathbf{C} \cup \{\text{dep}\}$ the set of all locations in the problem;
dem_{i,k}	the demand of commodity $k \in \mathbf{K}$ by customer $i \in \mathbf{C}$, we also denote by \mathbf{dem}_i the total demand of customer i (i.e. $\mathbf{dem}_i = \sum_{k \in \mathbf{K}} \mathbf{dem}_{i,k}$);
comp_{t,k}	the compatibility between vehicle type $t \in \mathbf{T}$ and commodity $k \in \mathbf{K}$
	$\mathbf{comp}_{t,k} = \begin{cases} 1 & \text{if vehicle type } t \text{ can transport commodity } k, \\ 0 & \text{otherwise;} \end{cases}$
E	the connections $e_{i,j}$, $\forall i, j \in \mathbf{C}'$ between the locations; and
cost_{t,i,j}	the cost of travelling on a connection $e_{i,j}$ with a vehicle of type t .
dist_{i,j}	the length of edge $e_{i,j}$.

We consider the distribution network to be a graph $\mathbf{G} = \langle \mathbf{C}', \mathbf{E} \rangle$ where the locations \mathbf{C}' represent nodes and the connections \mathbf{E} represent the edges. Moreover, all the models will refer to a set of vehicles $v \in \mathbf{V}$ that can be either fixed or to be decided. We list the different entities used to model the problem in ??.

3.2 Objective

The problem we address is the daily component of an annual fleet size-and-mix problem. The long term costs of running vehicles can be amortised to a daily fixed cost, but we decided not to take them into account as: *a)* our partenr was not able to provide those costs; *b)* they make little sense in the daily setting, although they are an easy extension.

Therefore the objective in our models is only to minimise the routing costs.

3.3 Constraints

We distinguish four types of *core* constraints that will be present in every problem variant.

Demand satisfaction the demands of all customers must be satisfied, that is the sum of all deliveries of a given commodity to a customer must equal its demand.

Capacity constraints the capacities of vehicles must be respected, that is the sum of all deliveries by a single vehicle must not exceed its capacity.

Used vehicles unused vehicles stay at the depot, and therefore cannot travel on any edge or serve any customers.

Visited customers vehicles can only deliver customers that they visited.

4 Our Approach

In this section we present our approach, whose modularity emerges from the natural decomposition of the addressed problem into fleet design and routing components. In particular, we describe two models for each component, which can be combined to assemble four different complete models. Moreover, we present a number of valid cuts and symmetry breaking constraints that can be applied to the models.

4.1 Common Variables

Our models share three sets of variables. The routing decisions are modelled using the Boolean variables x indicating whether a vehicle travels on a given edge in the solution. The amount of each commodity delivered by a vehicle at a customer is modelled by the continuous variables y . Finally, the Boolean variables u represent the *unused* vehicles, i.e., if u is equal to one, it means that the vehicle is not used.

The usage variable comes from the yearly problem and tells whether a vehicle is used on a given day. We decided to keep it in the daily setting as we found it

lead to stronger relaxations when deciding the fleet composition, it is also used to formulate some valid cuts (e.g., (??)).

4.2 Fleet Models

In this section we describe two different models to deal with the fleet design component of the problem. In the *Flexible* fleet, we have a fixed number of trucks and use a Boolean variable z to determine the type of each vehicle. We can easily compute the maximum size of the fleet by considering only the smallest trucks, then the question of the fleet composition remains. In the *Stable* fleet, we assign each vehicle a type beforehand and size the fleet according to their respective characteristics.

4.2.1 Flexible

In addition to the core variables this model uses an additional set of Boolean variables $z_{v,t}$ to indicate if vehicle $v \in \mathbf{V}$ is of type $t \in \mathbf{T}$. In order to keep a linear model, as different vehicle types have different costs, we have to decompose the routing decision on the type of the vehicle and therefore use variables $x_{v,t,i,j}$ to indicate if a vehicle $v \in \mathbf{V}$ of type $t \in \mathbf{T}$ travels on edge $e_{i,j} \in \mathbf{E}$. As such, vehicles have access to networks of different costs, but can only use the network of their chosen type. Finally, we need to enforce compatibility because the type of each vehicle is not known in advance.

$$\sum_{i \in \mathbf{C}} y_{v,i,k} \leq \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \cdot \mathbf{comp}_{t,k} \quad \forall v \in \mathbf{V}, k \in \mathbf{K} \quad (1)$$

$$\sum_{t \in \mathbf{T}} z_{v,t} = 1 \quad \forall v \in \mathbf{V} \quad (2)$$

$$\sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \leq |\mathbf{E}| \cdot z_{v,t} \quad \forall v \in \mathbf{V}, t \in \mathbf{T} \quad (3)$$

- (1) Compatibility constraint.
- (2) All vehicles must have exactly one type.
- (3) Vehicles can only use edges associated with their chosen type.

4.2.2 Stable

Instead of determining the composition of the fleet during the optimisation another approach is to have vehicles with fixed types from the start. We obtain a model with *pools* of vehicles of a given type. We then need to size each of these pools so that we are able to satisfy the demands.

Presetting the type of the vehicles allows us to use a three-index routing variable, $x_{v,i,j}$, thus reducing greatly the number of variables in the model.

Furthermore, we reduce the number of y variables as compatibility becomes an implicit characteristic. As it also eliminates infeasible deliveries, the compatibility constraints can be removed. In the following we denote \mathbf{K}^v the set of compatible commodities for vehicle v . Finally, the y variables are bounded by the size of the vehicle instead of the maximum size in the fleet.

4.3 Routing Models

In this section we present two different models to deal with the routing aspects of the problem. Both are based on classic formulations in the literature. The first model is called *Vehicle Flow*, reflecting the fact that the main constraint models the balance in the in-flow and out-flow from a node. The second model is called *Commodity Flow*, since it employs an additional variable tracking the amount of goods carried on the edges.

4.3.1 Vehicle Flow

This model is very close to the standard 2-index model from the literature Laporte and Nobert (1983), and requires a rather small number of variables. The decision variables are the amount of flow on each vehicle between each pair of customers.

$$\sum_{j \in \mathbf{C}'} x_{v,i,j} = \sum_{j \in \mathbf{C}'} x_{v,j,i} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}' \quad (4)$$

This formulation does not automatically eliminate sub-tours, and hence sub-tour elimination constraints must be included. Because these are exponential in the number of nodes, we use a classic sub-tour elimination procedure, which fires whenever an integer solution is found during the branch-and-bound search. If we find any sub-tour S not including the depot, the following constraint is added, with \mathbf{E}^S being the set of edges in S :

$$\sum_{e_{i,j} \in \mathbf{E}^S} x_{v,i,j} \leq |S| - 1 \quad \forall S \subseteq \mathbf{C}, v \in \mathbf{V} \quad (5)$$

4.3.2 Commodity Flow

This model uses a set of variables $f_{v,k,i,j}$ representing the amount of commodity k transported by vehicle v on $e_{i,j}$ (see Gavish and Graves, 1978). As a consequence, the Commodity flow formulation has more variables than the Vehicle flow formulation but results in a polynomial-sized model since no sub-tour elimination constraints are required. In order to guarantee flow consistency, we need to add the following set of constraints:

$$\sum_{j \in \mathbf{C}'} (f_{v,k,j,i} - f_{v,k,i,j}) = y_{v,i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K} \quad (6)$$

$$\sum_{k \in \mathbf{K}} f_{v,k,i,j} \leq x_{v,i,j} \cdot \mathbf{cap}_v \quad \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} \quad (7)$$

- (6) The amount deposited at i by vehicle v is the difference between the amount carried before the visit and after the visit.
- (7) A vehicle can only carry a load on the edges it uses.

4.4 Valid Cuts

Valid cuts (or *inequalities*) are an extremely effective technique to solve MIP problems efficiently. Similarly to *redundant constraints* in CP, they are used to improve the strength of currently existing constraints without removing optimal solutions. In this section we will present a set of valid cuts derived from the literature and adapted to work in our rich VRP context.

We do not present all variants of the constraints, only their most generic expression. If a constraint differs considerably between two models, we present the two formulations; otherwise, assume no changes apart from the indices.

4.4.1 Minimum Visits

This cut (Dror et al, 1994) provides a lower bound on the number of vehicles required to visit a given customer, it is obtained by dividing its total demand by the maximum capacity in our fleet.

$$\sum_{v \in \mathbf{V}} \sum_{j \in \mathbf{C}'} x_{v,j,i} \geq \left\lceil \frac{\mathbf{dem}_i}{\mathbf{cap}_{max}} \right\rceil \quad \forall i \in \mathbf{C} \quad (8)$$

4.4.2 Minimum Vehicles

This constraint, together with the branching strategy, was the most successful improvement in the search for feasible solutions with a flexible fleet in Kilby and Urli (2016).

In the Vehicle model a similar cut can be implemented by ensuring that we will have enough vehicle capacity to serve all demands.

$$\sum_{v \in \mathbf{V}} \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \geq \sum_{i \in \mathbf{C}} \mathbf{dem}_{i,k} \quad \forall k \in \mathbf{K} \quad (9)$$

In the Stable model, since capacities are already fixed, we simply ensure that enough vehicles leave the depot by using the u (usage) variables. This provides an earlier cut in the branch-and-bound search since us have high branching priority. Let F_t be the total number of vehicle of type t .

$$\sum_{v \in \mathbf{V}} u_v \leq \sum_{t \in \mathbf{T}} F_t - \sum_{i \in \mathbf{C}} \left\lceil \frac{\mathbf{dem}_i}{\mathbf{cap}_{max}} \right\rceil \quad (??')$$

4.4.3 Maximum Vehicles

Conversely, when using the Stable fleet model, we can limit the number of vehicle in operation by imposing a lower bound on the number V^{dep} of vehicles staying at the depot.

$$\sum_{v \in \mathbf{V}} u_v \geq V^{\text{dep}} \quad (10)$$

4.4.4 Fractional Sub-Tours

This cut (Dror et al, 1994) requires that if a vehicle visits a customer it has to travel somewhere else afterwards.

$$x_{v,i,j} \leq \sum_{l \in \mathbf{C}', l \neq i} x_{v,j,l} \quad \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} \quad (11)$$

4.4.5 Depot Outgoing Degree

In its general form, this cut (Dror et al, 1994) states that all vehicles have to exit the depot, however we amend it to allow unused vehicles to stay at the depot.

$$\sum_{j \in \mathbf{C}} x_{v,\text{dep},j} + u_v = 1 \quad \forall v \in \mathbf{V} \quad (12)$$

4.4.6 Single visit.

This cut enforces the fact that a vehicle can visit a customer at most once, using the following symmetric constraints:

$$\sum_{j \in \mathbf{C}'} x_{v,i,j} \leq 1 \quad \forall v \in \mathbf{V}, i \in \mathbf{C} \quad (13)$$

$$\sum_{j \in \mathbf{C}'} x_{v,j,i} \leq 1 \quad \forall v \in \mathbf{V}, i \in \mathbf{C} \quad (14)$$

4.5 Symmetry Breaking Constraints

Symmetry breaking constraints try to alleviate the degeneracies of a model by removing equivalent solutions. In this section we assume that the input data, such as vehicles or customers, is ordered as integer sequences. In particular, the depot would be customer 0.

4.5.1 Usage

If a vehicle is unused, the *next* vehicle must be unused too so as to avoid recombination to determine the number of used vehicles.

$$u_{v-1} \geq u_v \quad \forall v \in \mathbf{V} \quad (15)$$

4.5.2 Visit Order

We impose that a vehicle v can visit customer i only if vehicle $v - 1$ has visited any location in $[1..i]$.

$$\sum_{h \in \mathbf{C}'} \sum_{l=1}^j x_{v-1,h,l} \geq x_{v,i,j} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, j \in \mathbf{C}' \quad (16)$$

In the Flexible model this must be restricted to vehicles of the same type.

$$\sum_{u=0}^t z_{v-1,u} + \sum_{h \in \mathbf{C}} \sum_{l=1}^j x_{v-1,t,h,l} \geq x_{v,t,i,j} \quad \forall v \in \mathbf{V}, t \in \mathbf{T}, i \in \mathbf{C}, j \in \mathbf{C}' \quad (??')$$

4.5.3 Fleet Order

When using the Flexible model, we force the vehicle types to be ordered, that is if vehicle v is of type t , then vehicle $v+1$ can only take types $[t..T]$.

$$\sum_{\tau=0}^t z_{v-1,\tau} \geq z_{v,t} \quad \forall v \in \mathbf{V}, t \in \mathbf{T} \quad (17)$$

4.5.4 Customer Assignment

Inspired by Dror et al (1994), where the authors design a cut assigning the first vehicle to the farthest customer, we re-order the input data so that the farther customers (from the depot) are assigned smaller indices.

Using a Flexible fleet, with \hat{j} being the farthest customer, we have:

$$\sum_{i \in \mathbf{C}} \sum_{t \in \mathbf{T}} x_{0,t,i,\hat{j}} = 1 \quad (18)$$

With a Stable fleet we can only have a weaker constraint where we force at least one of the first vehicles of each type to visit the farthest customer. With Π being the sets of such vehicles.

$$\sum_{v \in \Pi} \sum_{i \in \mathbf{C}} x_{v,i,\hat{j}} \geq 1 \quad (??')$$

4.5.5 Total Load

In the Commodity flow model we also enforce that all vehicles have to leave the depot fully loaded. However, we do not constrain their final load as this can be adjusted easily.

$$\sum_{k \in \mathbf{K}} \sum_{j \in i} f_{v,k,\text{dep},j} = \mathbf{cap}_v \quad \forall v \in \mathbf{V} \quad (19)$$

4.6 Complete Example

We present here a complete mathematical formulation using: Commodity routing model (?????) and Stable fleet model (e.g. implicit compatibility with \mathbf{K}^v). The model is composed of: the objective function, core constraints (?????????), valid cuts (?????????????????), symmetry breaking constraints (????), and ordering constraints (??).

$$\begin{aligned}
\min \quad & \sum_{v \in \mathbf{V}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \cdot \mathbf{cost}_v \cdot \mathbf{dist}_{i,j} \quad (\text{MIP}) \\
s.t. \quad & \sum_{v \in \mathbf{V}} y_{v,i,k} = \mathbf{dem}_{i,k} \quad \forall i \in \mathbf{C}, k \in \mathbf{K} \quad (20a) \\
& \sum_{i \in \mathbf{C}} \sum_{k \in \mathbf{K}^v} y_{v,i,k} \leq \mathbf{cap}_v \quad \forall v \in \mathbf{V} \quad (20b) \\
& \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \leq u_v \quad \forall v \in \mathbf{V} \quad (20c) \\
& \sum_{j \in \mathbf{C}} x_{v,j,i} \cdot \mathbf{dem}_{i,k} \geq y_{v,i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K}^v \quad (20d) \\
& \sum_{j \in \mathbf{C}'} (f_{v,k,j,i} - f_{v,k,i,j}) = y_{v,i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K}^v \quad (20e) \\
& \sum_{k \in \mathbf{K}^v} f_{v,k,i,j} \leq x_{v,i,j} \cdot \mathbf{cap}_v \quad \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} \quad (20f) \\
& \sum_{v \in \mathbf{V}} \sum_{j \in \mathbf{C}'} x_{v,j,i} \geq \left\lceil \frac{\mathbf{dem}_i}{\mathbf{cap}_{max}} \right\rceil \quad \forall i \in \mathbf{C} \quad (20g) \\
& \sum_{v \in \mathbf{V}} u_v \leq \sum_{t \in \mathbf{T}} F_t - \sum_{i \in \mathbf{C}} \left\lceil \frac{\mathbf{dem}_i}{\mathbf{cap}_{max}} \right\rceil \quad (20h) \\
& \sum_{v \in \mathbf{V}} u_v \geq V^{\mathbf{dep}} \quad (20i) \\
& x_{v,i,j} \leq \sum_{l \in \mathbf{C}', l \neq i} x_{v,j,l} \quad \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} \quad (20j) \\
& \sum_{j \in \mathbf{C}} x_{v,\mathbf{dep},j} + u_v = 1 \quad \forall v \in \mathbf{V} \quad (20k) \\
& \sum_{j \in \mathbf{C}'} \sum_{k \in \mathbf{K}^v} f_{v,k,\mathbf{dep},j} = \mathbf{cap}_v \quad \forall v \in \mathbf{V} \quad (20l) \\
& \sum_{j \in \mathbf{C}'} x_{v,i,j} \leq 1 \quad \forall v \in \mathbf{V}, i \in \mathbf{C} \quad (20m) \\
& \sum_{j \in \mathbf{C}'} x_{v,j,i} \leq 1 \quad \forall v \in \mathbf{V}, i \in \mathbf{C} \quad (20n) \\
& u_{v-1} \geq u_v \quad \forall v \in \mathbf{V} \quad (20o) \\
& \sum_{h \in \mathbf{C}'} \sum_{l=1}^j x_{v-1,h,l} \geq x_{v,i,j} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, j \in \mathbf{C}' \quad (20p) \\
& \sum_{v \in \Pi} \sum_{i \in \mathbf{C}} x_{v,i,j} \geq 1 \quad (20q) \\
& x \in \mathbb{B}, u \in \mathbb{B}, y \geq 0, f \geq 0
\end{aligned}$$

5 Experimental Comparison

In this section we compare several variants of the presented models. Each experiment is run for 15 minutes on an AMD Opteron 4334 machine running at 3.1 GHz and with 64 GB of RAM. The models were implemented in GUROBI 6.5 (Gurobi Optimization, 2015), and run using the default parameters except a higher branching priority for the usage variable for all models, and then for the truck type in the Flexible variant.

We have three groups of instances: small, medium, and large. The latter represent real demand data, whereas in medium and small a different degree of clustering is used to aggregate the demands of nearby customers. On average, there are 5 customer nodes in a small instance, 15 in a medium, and 18 in a large.

To size the fleet, we rely on a characteristic of our case study: refrigerated vehicles are more expensive to run than regular vehicles. Hence, we only consider chilled demand for determining the number of refrigerated vehicles required, and ambient demand for regular vehicles.

In our results we use the following abbreviations: the first letter is for the fleet model (s for Stable and f for Flexible); the second is for the routing model (c for Commodity and v for Vehicle).

5.1 MIP Comparison

This first set of experiments reports the results obtained over nine small instances by the four MIP models, augmented with valid cuts, symmetry constraints, and ordering constraints. Each instance was run for a maximum of fifteen minutes. We report the best objective value found across all variants; then the run time in seconds (if the time limit was not exceeded) or the final integrality gap (if the time limit was hit) for each of the variants; and finally the number of sub-tour elimination constraints (??) added in the two models based on Vehicle flow. The last line recaps the average gap found on instances not solved to optimality. We highlight rows where optimality was not reached by any model as well as the best time or gap per row.

The tables in ?? provide the results for the complete set of experiments, starting with the standard MIP formulation as presented in Kilby and Urli (2016). ?? presents the results of the most complete model which consists of: valid cuts, symmetry breaking constraints, and ordering.

In ?? we provide the time taken in seconds and the gap between the optimal solution, or best known solution, and the linear relaxation; and respectively with the first heuristic solution in ??.

From these results we can clearly see that a model dominates the others: the Stable fleet with Commodity routing, Model SC, presented in its augmented version as Model (20). First of all, predefining the types of each vehicle instead of having to determine their type during the optimization is the biggest improvement. While it increases the number of vehicles to route it decreases the number of decision variables. Secondly, having a polynomial-sized formulation by using a flow-based formulation for the routing gives far better results than relying on sub-tour elimination. This is explained by the fact that such a formulation has a better linear relaxation (see ??), and therefore a smaller search tree. Conversely,

Table 1 Results of using different models. For each instance we report the best value found, time taken or integrality gap at 15 min, and number of sub-tour elimination constraints. The best results are in bold. Instances where the optimum is not reached are greyed.

Instance	Value	Time (s) or gap at 900s				# subtours (??)	
		sc	sv	fc	fv	sv	fv
1	31358.95	15.19	26.02	32.63	33.81	216	250
2	45538.02	723.32	758.51	(0.20%)	(1.13%)	196	325
3	54150.16	70.89	343.70	503.72	(0.62%)	456	1020
4	45435.52	326.36	680.91	505.64	(1.30%)	341	490
5	53989.22	817.45	(1.02%)	(1.27%)	(1.79%)	117	255
6	60943.93	(0.11%)	(1.55%)	(0.83%)	(3.13%)	765	1700
7	62864.21	(0.62%)	(1.81%)	(1.12%)	(3.77%)	430	1700
8	23219.89	2.25	2.35	25.46	22.99	0	0
9	44738.91	125.65	259.11	281.49	550.84	203	140
Avg. gap		0.36%	1.46%	0.85%	1.96%		

Table 2 Root node relaxation results. The Time columns give the time taken to solve the relaxed problem; the Gap columns give the percentage difference with the best known value. The smallest gaps are in bold. Instances where optimum is not reached are greyed.

Instance	sc		sv		fc		fv	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
1	0.44	6.61	0.35	88.64	2.04	8.53	1.18	86.36
2	0.52	8.41	0.29	81.05	3.07	8.92	1.23	99.98
3	1.09	3.35	0.48	112.97	4.99	4.13	3.02	136.74
4	0.73	3.43	0.45	112.15	3.58	3.56	1.92	138.54
5	1.05	4.38	0.42	83.89	4.68	4.12	2.16	128.39
6	1.10	4.47	0.61	111.09	6.86	4.57	3.48	131.70
7	1.47	4.55	0.50	111.39	6.11	4.88	4.12	153.74
8	0.10	13.99	0.07	14.36	0.67	13.22	0.40	77.54
9	0.66	8.05	0.33	79.06	2.73	8.86	1.59	95.60
Avg. gap (%)		6.89		81.73		7.33		109.02

the reason why the first integer solution for flow based models is better is because they need to add sub-tour elimination constraints to reach feasibility, and it is reflected in the time they take to find that first solution.

This result concurs with those of Yaman (2006) whose best performing model is based on a flow model and fleet is always predefined.

5.2 Warm-Starting the MIP Using CP

In this second set of experiments we attempt to integrate CP to MIP. We devised a rather lightweight integration between the CP solver in Kilby and Urli (2016) and our MIP models. Since the CP-based solver returns a reasonably good solution quickly, the idea is: first, find a solution with CP; then use that solution as a starting solution for the MIP; and, from there, let the MIP search proceed normally. We call this: *warm starting* the MIP.

Table 3 First integer solution results. The Time columns give the time taken to find the first integer solution; the Gap columns the percentage difference with the best known value. The smallest gaps are in bold. Instances where optimum is not reached are greyed.

Instance	sc		sv		fc		fv	
	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)	Time (s)	Gap (%)
1	0.15	56.16	0.08	54.20	0.46	55.85	0.44	42.76
2	0.10	54.63	0.01	50.24	9.27	13.84	8.99	13.00
3	0.15	56.68	0.38	37.90	7.13	34.25	2.53	38.10
4	0.10	55.43	0.03	45.64	16.71	21.59	1.69	27.01
5	0.16	56.08	0.11	32.80	8.00	32.46	1.89	31.66
6	0.16	57.59	0.01	49.69	29.16	10.33	2.99	32.88
7	0.18	57.31	0.39	39.54	64.35	19.13	3.62	38.09
8	0.01	44.76	0.00	36.79	0.19	47.12	0.14	44.07
9	0.14	58.67	0.01	53.48	23.59	17.76	1.24	29.88
Avg. gap (%)	54.63		44.44		31.84		32.35	

In the following experiments, we keep the same 15-minutes time limit as before, but we allow a variable amount of time (ranging between 0 and 15 minutes) to the CP solver to find an initial solution. In order to keep the length of the experiments constant, the longer the CP search runs, the shorter the time limit for the MIP search.

The CP solver functions by finding a first feasible solution and then proceeding to improve it by means of an LNS procedure. Therefore, in our experiments, allocating a time of 0 seconds to the CP solver means we use the first feasible solution it finds instead of letting the CP solver improve it.

?? presents the evolution of the final MIP gap on a selected instance, day 6, for all instance sizes and model variations. We report the final integrality gap function of the time allocated to CP.

The slight dip early in ?? shows that the first solution found by CP is not the most efficient for warm starting the MIP search, and letting CP improve it for about one minute yields better results. Allowing longer CP run-times decreases the overall solution quality. The reason behind this is that CP finds a feasible solution quickly, then has to spend more and more time improving it as the solutions get better. At a given point, letting MIP improve the solution proves to be a better strategy than letting CP do it. For the flexible fleet with commodity flow, it is even impossible to find even the solution to the linear relaxation at the root node under fifteen minutes on large instances.

?? reports the results of warm-starting the MIP using a solution found by the CP model in 1 minute; it presents the results in a similar fashion to those in ???. Do note that the time reported is for the MIP only, not taking the CP runtime into account.

Warm starting the MIP search with a CP solution gives satisfying results: one more instance is now solved to optimality under the 15 minutes constraint; overall solving times are reduced; and, although the average gap reported on the last line is higher for some models – e.g. SC – this comes from having less unsolved days – and then the gap is lower, 0.62% vs. 0.45%.

On the other hand, we can observe a slight decrease in performance in some cases, with higher time or final gaps. This comes from the *starting location* for

Fig. 1 Evolution of final gap when using CP solution, based on a single instance, day 6, for the four models. The x axis is the time allotted to the CP model, then the MIP has the remaining of the fifteen minutes. A dot means the MIP finished. A missing curve means that the LP relaxation at the root node did not finish under the remaining time.

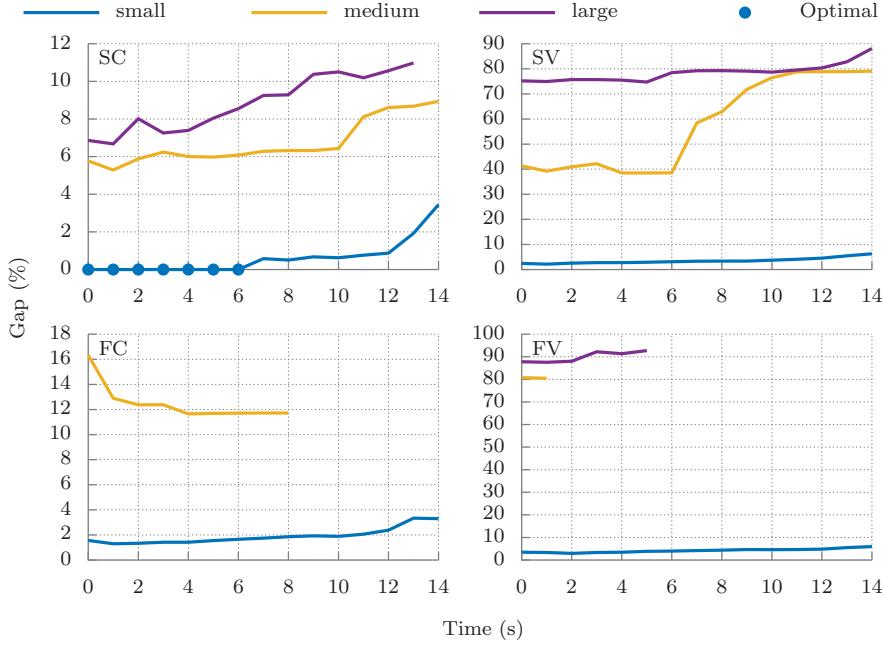


Table 4 Warm-starting MIP with a CP solution found in 1 minute

Instance	Value	Time (s) or gap at 900s				# subtours (??)	
		sc	sf	fc	ff	sf	ff
1	31358.95	29.31	27.82	19.83	58.84	120	300
2	45538.02	522.84	806.01	598.22	(0.36%)	196	260
3	54150.16	72.99	664.07	506.86	(0.26%)	342	340
4	45435.52	222.91	207.41	438.87	517.95	62	770
5	53989.22	447.78	(1.48%)	(1.37%)	(1.65%)	78	425
6	60943.93	664.57	(2.15%)	(1.10%)	(3.51%)	270	800
7	62864.21	(0.45%)	(1.09%)	(1.53%)	(3.16%)	301	1200
8	23219.89	4.58	4.61	27.21	18.78	36	40
9	44738.91	160.73	183.37	161.06	438.17	261	770
Avg. gap		0.45%	1.58%	1.34%	1.79%		

the MIP; in other words, imposing a starting solution to the MIP model makes the search start at a given position in the tree, with no warranty that the optimal solution is close. In that case, letting the MIP model do the exploration and pruning may lead to a smaller search tree as the region of the initial solution will never be explored.

We can observe that most model find the optimal solution early on but fail to prove its optimality, this is shown by the best value reported for unsolved instances

being equal to the optimal value. Using the CP to warm start the search helps reducing the size of the search tree, but not always. A way to improve the solving process would be to determine which solutions are interesting, or find a way to close the integrality gap faster.

5.3 Results Over a Year

In the final experiment from Kilby and Urli (2016, Table 2) the authors compared the results of two CP models, one with a standard B&B search and one with the custom LNS search, and their MIP model using one year of real world demand provided by the client.

We ran the same experiment using the MIP model (20). ?? summarizes the results on all instances, grouped by quarter, of: the CP model using LNS search, our MIP, and our MIP warm-started with a CP solution obtained in one minute.

For each quarter, we report the average value found across all the instances, how often the model produced the best known solution, the percentage difference with the best found value for each instance, and the final gap for the MIP and the combination CP+MIP.

Table 5 Summary of results using one year of data, grouped by quarter, for each instance size. For each approach we have the average value over all instances, the percentage of instances where the approach found the best known value, and the percentage difference with the best known value. We also report the average integrality gap for the MIP based approaches.

CP			MIP				CP (1min) + MIP				
	Value	Best (%)	Diff. (%)	Value	Best (%)	Diff. (%)	Gap (%)	Value	Best (%)	Diff. (%)	Gap (%)
Q1	S 48959	0.00	0.78	48582	7.78	0.00	0.81	48582	92.22	0.02	0.82
	M 51353	7.78	3.59	51320	40.00	3.50	7.39	50068	52.22	0.98	6.03
	L 52406	6.67	4.96	52455	27.78	5.00	10.17	50456	65.56	1.33	7.46
Q2	S 47511	1.11	1.06	47030	7.78	0.01	0.68	47026	91.11	0.01	0.76
	M 49896	5.56	4.06	48954	47.78	2.42	6.50	48634	46.67	1.59	6.27
	L 50956	6.67	5.45	50602	27.78	4.43	9.89	48921	65.56	1.32	7.45
Q3	S 49043	0.00	1.04	48534	7.69	0.00	0.74	48532	92.31	0.01	0.80
	M 51359	5.49	3.72	50551	37.36	2.37	6.88	50062	57.14	1.61	6.20
	L 52532	8.79	5.19	52090	26.37	4.07	9.99	50652	64.84	1.75	8.02
Q4	S 51700	2.20	0.93	51261	12.09	0.00	0.73	51259	85.71	0.01	0.77
	M 54182	12.09	3.94	53972	30.77	2.88	7.12	53022	57.14	1.65	6.00
	L 54179	8.79	4.72	53654	23.08	3.17	9.34	52557	67.03	1.85	7.81
Y	S 49303	0.83	0.95	48852	8.83	0.00	0.74	48850	90.34	0.01	0.79
	M 51698	7.73	3.83	51199	38.98	2.79	6.97	50446	53.29	1.46	6.12
	L 52518	7.73	5.08	52200	26.25	4.17	9.85	50646	65.74	1.56	7.68

The combination CP+MIP consistently dominates its individual components. Even though the MIP sometimes reaches a better value in the end, especially on medium size instances, overall the combination CP+MIP has a lower value overall, as shown by the lower percentage difference overall. The combined models also have a lower integrality gap after 15 minutes. Furthermore, the combined

CP+MIP has a greater success rate in finding the optimal under the time limit (64%) compared to the simple MIP (48%).

Instances *better* solved using the MIP without warm-start are symptomatic of those cases where the CP solution, albeit of good quality, is *far* from the actual optimal and the MIP spends a lot of time backtracking in the search tree. In *small* instances, the MIP has a lower percentage difference and better gap because it solves less instances to optimality, therefore the averages are lower.

6 Conclusions

We presented various MIP formulations for the fleet size and mix, multi-commodity, split-delivery vehicle routing problem with compatibility constraints. Our formulation is modular, in that we split the fleet design and the routing components of the problem and we develop two models for each component.

We carried out an extensive experimental analysis to compare the MIP models, and to measure their performance with respect to the CP-based LNS approach presented in Kilby and Urli (2016). Overall, the Stable fleet model and Commodity routing model combination seem to attain the best performance, outperforming the LNS approach on most instances, and represents the most promising candidate for a further exploration of decomposition models for the multi-day extension of the problem. Within our experimental analysis, we have also explored a lightweight integration of CP and MIP, which consists in priming the MIP model with a solution found by CP, replacing the heuristic solution found by the standard MIP solver (in our case GUROBI 6.5). The effect on the integrality gap is positive, however enough time must be allotted to the MIP search procedure to find a better solution.

References

- Baldacci R, Mingozzi A (2009) A unified exact method for solving different classes of vehicle routing problems. Mathematical Programming 120(2):347–380, DOI 10.1007/s10107-008-0218-9
- Baldacci R, Battarra M, Vigo D (2008) Routing a heterogeneous fleet of vehicles. Operations Research/ Computer Science Interfaces Series 43(1959):3–27, DOI 10.1007/978-0-387-77778-8{_}1
- Baldacci R, Mingozzi A, Roberti R (2012) Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. European Journal of Operational Research 218(1):1–6, DOI 10.1016/j.ejor.2011.07.037
- Caceres-Cruz J, Arias P, Guimaraes D, Riera D, Juan AA (2014) Rich vehicle routing problem: Survey. ACM Computing Surveys (CSUR) 47(2):32, DOI 10.1145/2666003
- Clarke G, Wright JW (1964) Scheduling of vehicles from a central depot to a number of delivery points. Operations research 12(4):568–581, DOI 10.1287/opre.12.4.568
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. Management science 6(1):80–91, DOI 10.1287/mnsc.6.1.80

- Dror M, Trudeau P (1989) Split Delivery Routing. *Transportation Science* 23(2):141–145, DOI 10.1287/trsc.23.2.141
- Dror M, Laporte G, Trudeau P (1994) Vehicle routing with split deliveries. *Discrete Applied Mathematics* 50(3):239–254, DOI 10.1016/0166-218X(92)00172-I
- Gavish B, Graves SC (1978) The travelling salesman problem and related problems. Operations Research Center Working Paper OR 078-78
- Golden B, Assad A, Levy L, Gheysens F (1984) The fleet size and mix vehicle routing problem. *Computers & Operations Research* 11(1):49–66, DOI 10.1016/0305-0548(84)90007-8
- Gurobi Optimization I (2015) Gurobi optimizer reference manual. URL <http://www.gurobi.com>
- Jabali O, Gendreau M, Laporte G (2012) A continuous approximation model for the fleet composition problem. *Transportation Research Part B: Methodological* 46(10):1591–1606, DOI 10.1016/j.trb.2012.06.004
- Kilby P, Urli T (2016) Fleet design optimisation from historical data using constraint programming and large neighbourhood search. *Constraints* 21(1):2–21, DOI 10.1007/s10601-015-9203-0
- Koç C, Bektaş T, Jabali O, Laporte G (2016) Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research* 249(1):1–21, DOI 10.1016/j.ejor.2015.07.020
- Laporte G, Nobert Y (1983) A branch and bound algorithm for the capacitated vehicle routing problem. *Operations-Research-Spektrum* 5(2):77–85, DOI 10.1007/BF01720015
- Lenstra JK, Kan A (1981) Complexity of vehicle routing and scheduling problems. *Networks* 11(2):221–227, DOI 10.1002/net.3230110211
- Letchford AN, Salazar-González JJ (2006) Projection results for vehicle routing. *Mathematical Programming* 105(2-3):251–274, DOI 10.1007/s10107-005-0652-x
- Letchford AN, Salazar-González JJ (2015) Stronger multi-commodity flow formulations of the Capacitated Vehicle Routing Problem. *European Journal of Operational Research* 244(3):730–738, DOI 10.1016/j.ejor.2015.02.028
- Toth P, Vigo D (2001) An overview of vehicle routing problems. In: *The vehicle routing problem*, Society for Industrial and Applied Mathematics, pp 1–26, DOI 10.1137/1.9780898718515.ch1
- Vidal T, Crainic TG, Gendreau M, Prins C (2014) A unified solution framework for multi-attribute vehicle routing problems. *European Journal of Operational Research* 234(3):658–673, DOI 0.1016/j.ejor.2013.09.045
- Yaman H (2006) Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming* 106(2):365–390, DOI 10.1007/s10107-005-0611-6

A Models

In this section we will present the different models created, without additional valid cuts or symmetry breaking constraints.

Model 1: Stable Fleet and Commodity Flow

$$\begin{aligned}
 \min & \sum_{v \in \mathbf{V}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \cdot \mathbf{cost}_v \cdot \mathbf{dist}_{i,j} && (\text{SC}) \\
 \text{s.t.} & \sum_{v \in \mathbf{V}} y_{v,i,k} = \mathbf{dem}_{i,k} && \forall i \in \mathbf{C}, k \in \mathbf{K} && (21\text{a}) \\
 & \sum_{i \in \mathbf{C}} \sum_{k \in \mathbf{K}^v} y_{v,i,k} \leq \mathbf{cap}_v && \forall v \in \mathbf{V} && (21\text{b}) \\
 & \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \leq u_v && \forall v \in \mathbf{V} && (21\text{c}) \\
 & y_{v,i,k} \leq \sum_{j \in \mathbf{C}} x_{v,j,i} \cdot \mathbf{dem}_{i,k} && \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K}^v && (21\text{d}) \\
 & \sum_{j \in \mathbf{C}'} (f_{v,k,j,i} - f_{v,k,i,j}) = y_{v,i,k} && \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K}^v && (21\text{e}) \\
 & \sum_{k \in \mathbf{K}^v} f_{v,k,i,j} \leq x_{v,i,j} \cdot \mathbf{cap}_v && \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} && (21\text{f}) \\
 & x \in \mathbb{B}, u \in \mathbb{B}, y \geq 0, f \geq 0
 \end{aligned}$$

Model 2: Stable Fleet and Vehicle Flow

$$\begin{aligned}
 \min & \sum_{v \in \mathbf{V}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \cdot \mathbf{cost}_v \cdot \mathbf{dist}_{i,j} && (\text{SF}) \\
 \text{s.t.} & \sum_{v \in \mathbf{V}} y_{v,i,k} = \mathbf{dem}_{i,k} && \forall i \in \mathbf{C}, k \in \mathbf{K} && (22\text{a}) \\
 & \sum_{i \in \mathbf{C}} \sum_{k \in \mathbf{K}^v} y_{v,i,k} \leq \mathbf{cap}_v && \forall v \in \mathbf{V} && (22\text{b}) \\
 & \sum_{e_{i,j} \in \mathbf{E}} x_{v,i,j} \leq u_v && \forall v \in \mathbf{V} && (22\text{c}) \\
 & y_{v,i,k} \leq \sum_{j \in \mathbf{C}} x_{v,j,i} \cdot \mathbf{dem}_{i,k} && \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K}^v && (22\text{d}) \\
 & \sum_{j \in \mathbf{C}'} x_{v,i,j} = \sum_{j \in \mathbf{C}'} x_{v,j,i} && \forall v \in \mathbf{V}, i \in \mathbf{C}' && (22\text{e}) \\
 & \sum_{e_{i,j} \in \mathbf{E}^S} x_{v,i,j} \leq |S| - 1 && \forall S \subseteq \mathbf{C}, v \in \mathbf{V} && (22\text{f}) \\
 & x \in \mathbb{B}, u \in \mathbb{B}, y \geq 0
 \end{aligned}$$

Model 3: Flexible Fleet and Commodity Flow

$$\begin{aligned}
\min \quad & \sum_{v \in \mathbf{V}} \sum_{t \in \mathbf{T}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \cdot \mathbf{cost}_t \cdot \mathbf{dist}_{i,j} && (\text{FC}) \\
s.t. \quad & \sum_{v \in \mathbf{V}} y_{v,i,k} = \mathbf{dem}_{i,k} \quad \forall i \in \mathbf{C}, k \in \mathbf{K} && (23a) \\
& \sum_{i \in \mathbf{C}} \sum_{k \in \mathbf{K}} y_{v,i,k} \leq \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \quad \forall v \in \mathbf{V} && (23b) \\
& \sum_{i \in \mathbf{C}} y_{v,i,k} \leq \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \cdot \mathbf{comp}_{t,k} \quad \forall v \in \mathbf{V}, k \in \mathbf{K} && (23c) \\
& \sum_{t \in \mathbf{T}} z_{v,t} = 1 \quad \forall v \in \mathbf{V} && (23d) \\
& \sum_{t \in \mathbf{T}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \leq u_v \quad \forall v \in \mathbf{V} && (23e) \\
& \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \leq |\mathbf{E}| \cdot z_{v,t} \quad \forall v \in \mathbf{V}, t \in \mathbf{T} && (23f) \\
& y_{v,i,k} \leq \sum_{t \in \mathbf{T}} \sum_{j \in \mathbf{C}'} x_{v,t,j,i} \cdot \mathbf{dem}_{i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K} && (23g) \\
& \sum_{j \in \mathbf{C}'} (f_{v,k,j,i} - f_{v,k,i,j}) = y_{v,i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K} && (23h) \\
& \sum_{k \in \mathbf{K}} f_{v,k,i,j} \leq x_{v,t,i,j} \cdot \mathbf{cap}_t \quad \forall v \in \mathbf{V}, e_{i,j} \in \mathbf{E} && (23i) \\
& x \in \mathbb{B}, u \in \mathbb{B}, y \geq 0, f \geq 0
\end{aligned}$$

Model 4: Flexible Fleet and Vehicle Flow

$$\begin{aligned}
\min \quad & \sum_{v \in \mathbf{V}} \sum_{t \in \mathbf{T}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \cdot \mathbf{cost}_t \cdot \mathbf{dist}_{i,j} && (\text{FF}) \\
s.t. \quad & \sum_{v \in \mathbf{V}} y_{v,i,k} = \mathbf{dem}_{i,k} \quad \forall i \in \mathbf{C}, k \in \mathbf{K} && (24a) \\
& \sum_{i \in \mathbf{C}} \sum_{k \in \mathbf{K}} y_{v,i,k} \leq \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \quad \forall v \in \mathbf{V} && (24b) \\
& \sum_{i \in \mathbf{C}} y_{v,i,k} \leq \sum_{t \in \mathbf{T}} z_{v,t} \cdot \mathbf{cap}_t \cdot \mathbf{comp}_{t,k} \quad \forall v \in \mathbf{V}, k \in \mathbf{K} && (24c) \\
& \sum_{t \in \mathbf{T}} z_{v,t} = 1 \quad \forall v \in \mathbf{V} && (24d) \\
& \sum_{t \in \mathbf{T}} \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \leq u_v \quad \forall v \in \mathbf{V} && (24e) \\
& \sum_{e_{i,j} \in \mathbf{E}} x_{v,t,i,j} \leq |\mathbf{E}| \cdot z_{v,t} \quad \forall v \in \mathbf{V}, t \in \mathbf{T} && (24f) \\
& y_{v,i,k} \leq \sum_{t \in \mathbf{T}} \sum_{j \in \mathbf{C}'} x_{v,t,j,i} \cdot \mathbf{dem}_{i,k} \quad \forall v \in \mathbf{V}, i \in \mathbf{C}, k \in \mathbf{K} && (24g) \\
& \sum_{j \in \mathbf{C}'} x_{v,t,i,j} = \sum_{j \in \mathbf{C}'} x_{v,t,j,i} \quad \forall v \in \mathbf{V}, t \in \mathbf{T}, i \in \mathbf{C}' && (24h) \\
& \sum_{e_{i,j} \in \mathbf{E}^S} x_{v,t,i,j} \leq |S| - 1 \quad \forall S \subseteq \mathbf{C}, v \in \mathbf{V}, t \in \mathbf{T} && (24i) \\
& x \in \mathbb{B}, u \in \mathbb{B}, y \geq 0
\end{aligned}$$

B Result tables

In this section, we present intermediate results table highlighting the progression of the solution quality. First, the results based on the standard models described in the previous section; then the results with the addition of valid cuts; and finally the results when using both valid cuts and symmetry breaking constraints.

The tables are formatted similarly to ??: for each instance we report the best value found; for each model, the solving time, in seconds, or the gap, in percentage, if the time exceeds 15 min; for both Vehicle flow models, the number of sub-tour elimination constraints added.

Table 6 Standard MIP model

Instance	Value	Time (s) or gap at 900s				# subtours (??)	
		sc	sf	fc	ff	sf	ff
1	29458.79	(5.12%)	(66.38%)	(6.26%)	(78.87%)	1824	2700
2	40376.14	(5.75%)	(73.50%)	(7.59%)	(82.28%)	1764	3055
3	54150.16	(1.86%)	(77.03%)	(2.50%)	(86.83%)	5054	3825
4	45435.52	(1.63%)	(74.47%)	(3.90%)	(87.19%)	3379	4410
5	39206.94	(3.44%)	(60.47%)	(4.98%)	(80.77%)	2652	4930
6	52069.54	(2.56%)	(76.21%)	(4.48%)	(80.76%)	6210	8500
7	62864.21	(2.77%)	(75.58%)	(4.22%)	(86.06%)	5160	4900
8	23219.88	(8.43%)	(43.94%)	(10.03%)	(75.76%)	1044	1360
9	41724.50	(5.89%)	(71.85%)	(7.62%)	(78.96%)	2900	3010
Avg. gap		4.16%	68.82%	5.73%	81.94%		

Table 7 Summary: valid cuts

Instance	Value	Time (s) or gap at 900s				# subtours (??)	
		sc	sf	fc	ff	sf	ff
1	31358.95	(2.78%)	(4.47%)	(2.85%)	(4.70%)	384	350
2	45538.02	(2.97%)	(5.20%)	(1.65%)	(5.44%)	504	845
3	54150.16	(1.17%)	(6.69%)	(3.96%)	(33.38%)	1330	510
4	45435.52	(1.22%)	(2.94%)	(3.30%)	(21.11%)	744	420
5	54010.05	(3.16%)	(4.59%)	(3.85%)	(6.51%)	936	425
6	60964.35	(2.72%)	(4.93%)	(4.40%)	(45.90%)	405	1300
7	63045.96	(1.72%)	(4.62%)	(4.61%)	(40.51%)	258	1900
8	23219.89	252.93	258.06	(0.87%)	(4.15%)	0	160
9	44738.91	(4.78%)	(7.42%)	(7.32%)	(21.59%)	754	980
Avg. gap		2.57%	5.11%	3.64%	20.37%		

Table 8 MIP with valid cuts and symmetry breaking constraints

Instance	Value	Time (s) or gap at 900s				# subtours (??)	
		sc	sf	fc	ff	sf	ff
1	31358.95	36.75	55.17	132.64	126.41	384	250
2	45538.02	(0.38%)	(2.35%)	(0.98%)	(3.10%)	532	195
3	54150.16	284.99	(1.77%)	(0.32%)	(3.58%)	1140	425
4	45435.52	(0.11%)	524.05	(0.77%)	(1.24%)	372	630
5	53989.22	(1.16%)	(0.15%)	(1.39%)	(2.17%)	1053	680
6	60964.26	(0.25%)	(2.88%)	(1.39%)	(5.04%)	810	600
7	62865.43	(1.45%)	(2.09%)	(1.54%)	(4.29%)	559	500
8	23219.89	4.57	6.75	28.95	40.94	18	80
9	44738.91	(0.18%)	(0.99%)	(0.83%)	(0.85%)	464	210
Avg. gap		0.59%	1.70%	1.03%	2.89%		