2

Literature study

The vehicle routing problem is a generalization of the traveling salesman problem (TSP), which both are NP-hard [10]. In recent years, advances have been made regarding computation methods and an overview of exact solutions is given by Golden et al. [11]. The following chapter starts with an overview of the VRP and its extensions, some of which are utilized in this study. Then follows an overview of the mathematical formulations and methods to solve the variants of the problem.

2.1 Classifications and extensions to the VRP

If we allow for customer demands to be relatively large, one vehicle is not always sufficient to satisfy the demand of a single customer. In that case, it is reasonable to assume that multiple vehicles can visit the same customer or that one vehicle can visit the customer multiple times, resulting in extensions like split pick-ups or deliveries [12] or multi-trips [13]. In [12], Tavakkoli-Moghaddam et al. show that a split work between vehicles to meet customer demands can reduce both the total cost and the total time. Another common extension to the classical VRP is to include time windows (VRPTW) which implies that each customer has a given time interval in which the loading/unloading is allowed to be performed. These time windows can be either hard or soft, soft meaning that violating the time window only induces an extra penalty term in the cost function [14].

More advanced extensions related to the VRP can be found, for instance in Huang et al. [15], which describes the multi-trip VRPTW with pick-up demands for customers and with a capacitated unloading station in the depot, meaning that vehicles may need to wait until they are able to unload cargo at the depot. Huang et al. propose an exact solution with a branch-and-price algorithm; see Section 3.3.3 for further explanation. Another related study is made by Mo et al. [16], where customers are clustered and each cluster includes one self-pickup spot, in order to try to reduce the total supplier cost by avoiding a larger portion of last-mile deliveries.

A relatively up-to-date literature review of the EVRP is presented by Kucukoglu et al. [17], according to which the most frequent objective function components include the number of vehicles, their travel distance and travel time, rather than the electricity cost or the charging time. However, the electricity cost and charging time are often included in combination with other components, especially when time-related constraints are included. Related to this, it is originally assumed in the basic formulation of the EVRP that every vehicle fully charges when visiting a location with a charging station. Adding time-related constraints give rise to a new category of EVRPs with partial charging, different charging technologies, and battery swapping. Another important characteristic of EVRPs is how to compute or estimate the energy consumption. Kucukoglu et al. classify the energy consumption to be of either deterministic or stochastic nature. Deterministic energy consumption functions are further divided into linear and nonlinear functions, taking vehicle speed, cargo load weight, and road characteristics into consideration. Vehicle load capacities may differ (i.e., a heterogeneous fleet is used) but according to Kucukoglu et al., most existing studies of the EVRP concern a homogeneous fleet.

2.2 Problem formulations and solution methods

According to Koç et al. [3], VRPs usually are stated in either a *commodity flow* formulation or a *set partitioning/set covering* formulation. A commodity flow formulation can be briefly summarized as an explicit optimization model in which binary variables determines whether a vehicle travels between two locations or not. These formulations have relatively weak lower linear programming-bounds (LP-bounds, see Chapter 3), but studies, see e.g., Letchford and Salazar-Gonzalez [18], have been made to propose a stronger formulation of a multi-commodity flow with stronger lower LP-bounds.

Set partitioning/set covering models instead consider a set of known routes for the vehicles and determines which of these routes that should be used in order to obtain the minimum cost while still satisfying all constraints. An example of a set covering problem is given in Section 3.2.1. Depending on the requirements of the problem, the complicating part may be to find these routes, and that there could be a very large number of them. Preferably, one may try to find the routes with another, less constrained, VRP. Nevertheless, there exist multiple solution algorithms that thrive in or even require a variant of a set partitioning/covering formulation, such as column generation. In the following study, the problem will be stated first with a commodity flow formulation and later developed into a generalized set-covering formulation with resource constraints; see Chapter 4 for details.

There are many proposed solution methods for VRPs, such as exact methods, meta-heuristics and heuristics. A selection of heuristic methods include tabu search or heuristics connected to population search, see [19]. To obtain an exact solution, it is commonly proposed to use a branch-and-price algorithm, see Section 3.3.3 for explanation and [4] for an implementation. In [20], Konstantakopoulos et al. outline a classification of exact and approximating methods to solve VRPs. Kucukoglu et al. [17] conclude that most solution methods to EVRPs are heuristics due to the complexity of the problem. (Adaptive) Large Neighborhood Search or forward labeling algorithms with dynamic programming (see [5] for an implementation of both) are meta-heuristics or heuristics commonly used to solve EVRPs. Kucukoglu et al. [17] state that forward labeling algorithms often are more time-consuming compared to other heuristic methods, but typically yield better solutions.

2.3 Definitions of important terms

A selection of important terms is clarified below. The definitions are mostly equivalent to those made by Ghandriz et al. in [7].

Node	A location where it is possible to charge the vehicle battery and/or to pick up or deliver cargo. The depot, the customers and locations with a charging station compose all nodes in our problem.
Fleet	All vehicles that are used to satisfy the customer demands.
Service	Charging of vehicle battery and loading and unloading of cargo. When referring to service time, we will assume that charging and loading/unloading can be done simultaneously and consequently that the service time in a node equals the maximum of the charging time and the loading/unloading time.
Route	A sequence of nodes, starting and ending at the depot with at least one other node in between, but including each node at most once.
Trip	The act of driving on a route and perform necessary service in nodes belonging to that route.

2.4 Loading and unloading types

In our study, four loading/unloading types will be accounted for:

- Connecting or disconnecting an *additional semitrailer* (AST) to/from the vehicle
- Having an *on-board lift* (OBL) on the vehicle
- Using *on-board vehicle waiting* (OBW), i.e., to load/unload cargo with lift-trucks
- Using a *straddle carrier* (SC) in a node to lift containers to/from the vehicle

A more detailed overview are made by Ghandriz et al. in [21]. Each loading/unloading type has a unique corresponding time rate, seen in Appendix A, but apart from the time rates, multiple simplifications are made to reduce the complexity of the problem. The proposed model in Chapter 4 may therefore only serve as a model to be built upon to handle the loading/unloading complexity. Notable simplifications are found below:

- All amounts of cargo loaded/unloaded in a node are continuous variables
- Additional semitrailers, on-board lifts and straddle carriers already exist and consequently, using them do not induce a fixed cost
- The cost connected to the loading/unloading waiting time is the only running cost term for a specific loading/unloading type
- Inclusion of an on-board lift neither increases the vehicle weight nor reduces

the cargo capacity

A vehicle and a node can have multiple loading/unloading types. While the model proposed in Chapter 4 is computationally feasible using all four loading types with multiple allowed types for each vehicle and node, only OBL and OBW will be used for the presented results in Chapter 5 to reduce the impact from the simplifications above.

4 Mathematical formulations

The optimization model is initially formulated based on studies made by Ghandriz et al. [7] and Kindstrand and Nordgren [27]. The model is proposed based on the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, where $\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_\text{c} \cup \mathcal{V}_\text{s} = \{0, 1, \ldots\}$ denotes all nodes in the problem instance. All vehicles start and end at the depot, denoted by the set $\mathcal{V}_0 := \{0_-, 0_+\}$. The elements of the set represent the first and final visit to the depot, so observe that a vehicle that visits only $\{0_-\}$ and $\{0_+\}$ is in reality not traveling at all. The set $\mathcal{V}_\text{c}$ denotes the customer nodes and $\mathcal{V}_\text{s}$ denotes the nodes with a charging station. Observe that a node can be included in both $\mathcal{V}_\text{c}$ and $\mathcal{V}_\text{s}$ simultaneously. Nodes are linked pairwise with the directed arcs

$$\mathcal{A} = \{(i, j) | \, i, j \in \mathcal{V}, \, i \neq j, \, i \neq 0_+, \, j \neq 0_-\}, \tag{4.1}$$

where $(i, j) \in \mathcal{A}$ is the directed arc going from node i to node j.

Every node $i \in \mathcal{V}$ has a pick-up demand $D_{\text{L}i} \geq 0$ and a delivery demand $D_{\text{U}i} \geq 0$. Observe that demands are also defined in the depot and in nodes with only a charging station. However, we set $D_{\text{U},0_-} = D_{\text{L},0_+} = 0$ and $D_{\text{L}i} = D_{\text{U}i} = 0, i \in \mathcal{V}_\text{s} \setminus (\mathcal{V}_\text{c} \cap \mathcal{V}_0)$ so that nodes with only a charging station exclusively have zero demands and for the depot, it is only possible to pick-up cargo in the node $\{0_-\}$ while the deliveries are handled in $\{0_+\}$. For customers, at least one of the pick up demand and the delivery demand is strictly positive. All demands are satisfied with loading and unloading of cargo, which can be done by the loading types included in the set $\mathcal{H}$. These loading types have different associated processing times and every loading type is not applicable for every vehicle in every node.

The set $\mathcal{K} = \{1, 2, \ldots\}$ denotes the set of vehicles in the heterogeneous fleet, with varying limitations in battery capacity B^k, cargo mass capacity $C^{\text{m},k}$ and cargo volume capacity $C^{\text{V},k}$. The vehicles can perform a maximum of T trips and each vehicle has the common daily upper limit of working time $t^{\max}$. The sets, along with necessary variables and parameters to formulate the model, are presented in Tables 4.1, 4.2, and 4.3.

Table 4.1: All sets for the optimization models given in Sections 4.1 and 4.2.

$\mathcal{K}$	Vehicles

Continued on next page

Table 4.1: All sets for the optimization models given in Sections 4.1 and 4.2. (Continued)

$\mathcal{V}_0$	The depot
$\mathcal{V}_\mathrm{c}$	Customer nodes
$\mathcal{V}_\mathrm{s}$	Nodes with a charging station
$\mathcal{V} = \mathcal{V}_0 \cup \mathcal{V}_\mathrm{c} \cup \mathcal{V}_\mathrm{s}$	All nodes
$\mathcal{A}$	Arcs connecting the nodes
$\mathcal{H}$	Loading and unloading types
$\mathcal{T} = \{1, \dots, T\}$	Trips

Table 4.2: All variables for the optimization model given in Sections 4.1 and 4.2.

$x_{ij}^{k,t}$	$\begin{cases} 1, & \text{if vehicle } k \in \mathcal{K} \text{ travels on arc } (i,j) \in \mathcal{A} \text{ on trip } t \in \mathcal{T}, \\ 0, & \text{otherwise} \end{cases}$	
$y_{ij}^{k,t}$	Cargo mass for vehicle $k \in \mathcal{K}$ when travelling on arc $(i,j) \in \mathcal{A}$ for trip $t \in \mathcal{T}$	[kg]
$m_{\mathrm{L}i}^{k,h,t}$	Loaded cargo onto vehicle $k \in \mathcal{K}$, made with loading type $h \in \mathcal{H}$ in node $i \in \mathcal{V}$ during trip $t \in \mathcal{T}$	[kg]
$m_{\mathrm{U}i}^{k,h,t}$	Unloaded cargo from vehicle $k \in \mathcal{K}$, made with unloading type $h \in \mathcal{H}$ in node $i \in \mathcal{V}$ during trip $t \in \mathcal{T}$	[kg]
$z_{ij}^{k,t}$	Amount of electricity used by vehicle $k \in \mathcal{K}$ on arc $(i,j) \in \mathcal{A}$ during trip $t \in \mathcal{T}$	[kWh]
$f_{ij}^{k,t}$	Amount of electricity needed[2] for vehicle $k \in \mathcal{K}$ on arc $(i,j) \in \mathcal{A}$ on trip $t \in \mathcal{T}$	[kWh]
$q_i^{k,t}$	Battery level for vehicle $k \in \mathcal{K}$ upon arrival in node $i \in \mathcal{V}$ during trip $t \in \mathcal{T}$	[kWh]
$p_i^{k,t}$	Amount of charged electricity for vehicle $k \in \mathcal{K}$ in node $i \in \mathcal{V}$ during trip $t \in \mathcal{T}$	[kWh]
$\tau_i^{k,t}$	Arrival time to node $i \in \mathcal{V}$ for vehicle $k \in \mathcal{K}$ on trip $t \in \mathcal{T}$	[h]
$s_i^{k,t}$	Service time for vehicle $k \in \mathcal{K}$ in node $i \in \mathcal{V}$ during a trip $t \in \mathcal{T}$	[h]
α_i	$\begin{cases} 1, & \text{if a charging station is installed at node } i \in \mathcal{V} \setminus \mathcal{V}_\mathrm{s}, \\ 0, & \text{otherwise} \end{cases}$	
$w^{k,t}$	$\begin{cases} 1, & \text{if vehicle } k \in \mathcal{K} \text{ is used on trip } t \in \mathcal{T}, \\ 0, & \text{otherwise} \end{cases}$	

Continued on next page

Table 4.2: All variables for the optimization model given in Sections 4.1 and 4.2. (Continued)

$a_i^{k,t}$	$\begin{cases} 1, & \text{if loading is allowed in node } i \in \mathcal{V} \text{ for vehicle } k \in \mathcal{K} \text{ on trip } t \in \mathcal{T}, \\ 0, & \text{if unloading is allowed in node } i \in \mathcal{V} \text{ for vehicle } k \in \mathcal{K} \text{ on trip } t \in \mathcal{T} \end{cases}$

Table 4.3: Parameters for the optimization model given in Sections 4.1 and 4.2. Parameter values are found in Appendix A.

$C^{\mathrm{m},k}$	The cargo mass capacity for vehicle $k \in \mathcal{K}$	[kg]
$C^{\mathrm{V},k}$	The cargo volume capacity for vehicle $k \in \mathcal{K}$	[m^3]
B^k	The battery capacity for vehicle $k \in \mathcal{K}$	[kWh]
$D_{\mathrm{U}i}$	The delivery demand of cargo in node $i \in \mathcal{V}$	[kg]
$D_{\mathrm{L}i}$	The pick-up demand of cargo in node $i \in \mathcal{V}$	[kg]
$L_i^{h,k}$	$\begin{cases} 1, & \text{if loading and unloading with type } h \in \mathcal{H} \text{ is possible for vehicle } k \in \mathcal{K} \text{ in node } i \in \mathcal{V}, \\ 0, & \text{otherwise} \end{cases}$	
ρ	Density of cargo	[kg/m^3]
d_{ij}	Distance of arc $(i,j) \in \mathcal{A}$	[m]
F	A constant[3] used for calculating energy consumption	[kWh/(m·kg)]
G^k	A constant[3] used for calculating energy consumption of vehicle $k \in \mathcal{K}$	[kWh/m]
t_{ij}	Travel time on arc $(i,j) \in \mathcal{A}$	[h]
$t^{\mathrm{LU},h}$	Time required per cargo unit for loading/unloading type $h \in \mathcal{H}$	[h/kg]
t^{el}	Charging time per electricity unit	[h/kWh]
$t^{\max}$	Maximum number of working hours per day	[h]
c^{el}	Cost of electricity	[€/kWh]
c^{w}	Driver wage	[€/h]
c^{fix}	Installment cost of a charging station	[€]
c^{s}	Penalty cost of performing an extra trip	[€]
M	A large (enough) constant concerning electricity consumption	[kWh]

[2] Observe that $f_{ij}^{k,t}$ can be computed directly from the variable $y_{ij}^{k,t}$, according to the equality (4.2p), below.

[3] Both F and G^k are calculated from other parameters concerning the propulsion of the vehicle. However, the main part of the vehicle dynamics model and relevant parameter values are not included in this report.

4.2 Generalized set covering formulation

As VRPs and their extensions are combinatorial optimization problems, a column generation approach is taken to be able to solve larger instances in a reasonable computing time. Hence, we reformulate the model described in Section 4.1 as a *generalized* set covering problem. We define a route as a sequence of nodes with the depot only at the start and end of the sequence and at least one other node $i \in \mathcal{V} \setminus \mathcal{V}_0$ in between, but every node at most once. An example of a route could be the ordered set $\mathcal{S}_r = \{0_-, i_1, i_2, \ldots, i_n, 0_+\}$, where $n \geq 1$ and $i_k \neq i_l$ if $k \neq l$. A *feasible* route is a route that satisfies all constraints in the model (4.2) except (4.2f), (4.2g), (4.2j), (4.2k) and (4.2t), meaning that one route does not necessarily fulfill any customer demands and no trip ordering is considered. Instead, the demand constraints are considered in a "master" stage of the column generation algorithm. We also remove any possibility of adding charging stations if needed, i.e., $\alpha_i = 0$, $i \in \mathcal{V}$. We propose that this could be implemented in a more extensive problem that concerns both locating charging stations and managing vehicle fleets. Using this model for fleet management, the extensive problem can then be solved with for instance a Benders' decomposition technique [28].

Let $\mathcal{R}_k$ be a set of feasible routes for vehicle $k \in \mathcal{K}$. Every route $r \in \mathcal{R}_k$ has an associated time and cost, computed similarly as before with the first two terms in Equation (4.2a). For each feasible route $r \in \mathcal{R}_k$, we also know the amount of cargo loaded/unloaded with a specific type $h \in \mathcal{H}$ in every node $i \in \mathcal{S}_r$. If we now define the variables

$$\lambda^{k,r} \in \mathbb{N} = \{0, 1, \ldots\}, \quad r \in \mathcal{R}_k, \quad k \in \mathcal{K},$$

as the number of times vehicle k uses route r, i.e., the number of identical trips, and use the same notation for all other sets, parameters and sets as before, we conclude the generalized set covering formulation of the extended EFSMVRP as

$$\min_{\lambda} \quad \sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \sum_{i \in \mathcal{V}} \left(c^{\text{el}} \bar{p}_i^{k,r} + c^{\text{w}} \left(\sum_{j \in \mathcal{V}:(i,j) \in \mathcal{A}} t_{ij} \bar{x}_{ij}^{k,r} + \bar{s}_i^{k,r} \right) \right) \lambda^{k,r} \tag{4.3a}$$

subject to

$$\sum_{r \in \mathcal{R}_k} \sum_{i \in \mathcal{V}} \left(\bar{s}_i^{k,r} + \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{A}} t_{ij} \bar{x}_{ij}^{k,r} \right) \lambda^{k,r} \leq t^{\max}, \qquad k \in \mathcal{K}, \tag{4.3b}$$

$$\sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \sum_{h \in \mathcal{H}} \bar{m}_{\text{L}i}^{k,h,r} \lambda^{k,r} \geq D_{\text{L}i}, \qquad i \in \mathcal{V}_{\text{c}}, \tag{4.3c}$$

$$\sum_{k \in \mathcal{K}} \sum_{r \in \mathcal{R}_k} \sum_{h \in \mathcal{H}} \bar{m}_{\text{U}i}^{k,h,r} \lambda^{k,r} \geq D_{\text{U}i}, \qquad i \in \mathcal{V}_{\text{c}}, \tag{4.3d}$$

$$\lambda^{k,r} \in \mathbb{N}, \qquad r \in \mathcal{R}_k, \quad k \in \mathcal{K}. \tag{4.3e}$$

Denote this problem (EFSMVRP). Here, for each vehicle $k \in \mathcal{K}$ and route $r \in \mathcal{R}_k$, the values of all variables in model (4.2) are assumed to be constant parameters since we know which route we travel on, the travel time of it and how much we load-/unload and charge in each node. Hence, those variables are now denoted $\bar{x}, \bar{s}, \bar{m}_{\text{L}}$,

$\bar{m}_{\text{U}}$ and so on. This model has another major difference compared to model (4.2); we are no longer bounded by the upper limit of T trips. The objective function is similar to before, but with terms regarding creation of additional charging stations and trip ordering penalties removed. Constraints (4.3b), (4.3c) and (4.3d) assure that each vehicle obeys the permitted working time and that all customer demands are satisfied. Compared to the regular set covering problem (3.2), this problem is generalized since

- The coefficient matrix A is not restricted to be binary,
- The RHS is not restricted to be binary,
- Variables can take integer values.

Furthermore, we added constraints (4.3b) to ensure that every vehicle stays within permitted working time, resulting in a generalized set covering problem *with resource constraints.*

Note that the constraints (4.3c) and (4.3d) are inequality constraints, while (4.2j) and (4.2k) are equality constraints. Equality constraints reflect real life scenarios better, as customers may not be able to store more than their delivery demand or supply more than their pickup demand. However, solutions to a model consisting of inequality constraints (i.e., a set covering model) are more easily found than a model consisting of corresponding equality constraints (a set partitioning model) and thus we aim to obtain a good-enough solution hoping that the cost of transportation will minimize excessive cargo flow.

4.2.1 Creation of Master Problem and Subproblem

To perform column generation, we consider the restricted sets $\tilde{\mathcal{R}}_k \subseteq \mathcal{R}_k$, $k \in \mathcal{K}$, and formulate the Restricted Master Problem (RMP) as

$$\min_{\lambda} \quad \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{i\in\mathcal{V}} \left(c^{\text{el}} \bar{p}_i^{k,r} + c^{\text{w}} \left(\bar{s}_i^{k,r} + \sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} \bar{x}_{ij}^{k,r} \right) \right) \lambda^{k,r} \tag{4.4a}$$

subject to

$$\sum_{r\in\tilde{\mathcal{R}}_k} \sum_{i\in\mathcal{V}} \left(\bar{s}_i^{k,r} + \sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} \bar{x}_{ij}^{k,r} \right) \lambda^{k,r} \leq t^{\max}, \qquad k \in \mathcal{K}, \tag{4.4b}$$

$$\sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{h\in\mathcal{H}} \bar{m}_{\text{L}i}^{k,h,r} \lambda^{k,r} \geq D_{\text{L}i}, \qquad i \in \mathcal{V}_{\text{c}}, \tag{4.4c}$$

$$\sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{h\in\mathcal{H}} \bar{m}_{\text{U}i}^{k,h,r} \lambda^{k,r} \geq D_{\text{U}i}, \qquad i \in \mathcal{V}_{\text{c}}, \tag{4.4d}$$

$$\lambda^{k,r} \geq 0, \qquad r \in \tilde{\mathcal{R}}_k, \quad k \in \mathcal{K}. \tag{4.4e}$$

Note that constraints (4.4e) indicate that the integer restrictions of the variables $\lambda^{k,r}$ are relaxed, which is crucial as column generation is a method for linear programs. In the conclusion of the algorithm, the variables will however be set to be non-negative integers again and branch-and-bound is applied. The sets $\tilde{\mathcal{R}}_k$ are originally chosen to contain very few routes, selected in a way that a feasible solution still exists. Associated to the master problem is $|\mathcal{K}|$ subproblems, one for each vehicle. The idea of column generation is to find routes $r \in \mathcal{R}_k \setminus \tilde{\mathcal{R}}_k$ for each vehicle which can be added to our set of routes $\tilde{\mathcal{R}}_k$ and improve our solution. These routes are found by minimizing the reduced cost of the corresponding variables. As a negative reduced cost signals that adding the route will reduce the objective value (i.e., our cost), we choose to add routes with negative reduced cost until no negative reduced costs exist. Since there are many subproblems, it may happen that one subproblem yields improving routes at one stage even though no improving routes were generated from that subproblem at an earlier iteration of the algorithm. Thus, the algorithm stops when every subproblem yield non-negative reduced costs or when the solution of the master problem is not improved in a number of iterations; see Section 4.3 for more details.

We define

$$
\begin{aligned}
\bar{c}^{kr}_{\text{tot}} &:= \sum_{i \in \mathcal{V}} \left(c^{\text{el}} \bar{p}^{k,r}_i + c^{\text{w}} \left(\bar{s}^{k,r}_i + \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{A}} t_{ij} \bar{x}^{k,r}_{ij} \right) \right), \\
\bar{A}^{kr}_1 &:= \sum_{i \in \mathcal{V}} \left(\bar{s}^{k,r}_i + \sum_{j \in \mathcal{V}:(i,j) \in \mathcal{A}} t_{ij} \bar{x}^{k,r}_{ij} \right), \\
\bar{A}^{ikr}_2 &:= \sum_{h \in \mathcal{H}} \bar{m}^{k,h,r}_{\text{L}i}, \\
\bar{A}^{ikr}_3 &:= \sum_{h \in \mathcal{H}} \bar{m}^{k,r}_{\text{U}i},
\end{aligned}
$$

for each pair (k, r), where the values $\bar{p}^{k,r}_i$, $\bar{x}^{k,r}_{ij}$, $\bar{s}^{k,r}_i$, $\bar{m}^{k,h,r}_{\text{L}i}$ and $\bar{m}^{k,h,r}_{\text{U}i}$ are known for the route. Let μ_k, δ_i and γ_i be the linear programming dual variables corresponding to constraints (4.4b), (4.4c), and (4.4d) respectively. Then, the dual of (4.4) becomes

$$
\max_{\mu,\delta,\gamma} \quad \sum_{k \in \mathcal{K}} t^{\max} \mu_k + \sum_{i \in \mathcal{V}} \left(D_{\text{L}i} \delta_i + D_{\text{U}i} \gamma_i \right), \tag{4.5a}
$$

subject to

$$
\begin{aligned}
\bar{A}^{kr}_1 \mu_k + \sum_{i \in \mathcal{V}} \bar{A}^{ikr}_2 \delta_i + \sum_{i \in \mathcal{V}} \bar{A}^{ikr}_3 \gamma_i &\le \bar{c}^{kr}_{\text{tot}}, && r \in \mathcal{R}_k, \quad k \in \mathcal{K}, && (4.5\text{b}) \\
\mu_k &\le 0, && k \in \mathcal{K}, && (4.5\text{c}) \\
\delta_i &\ge 0, && i \in \mathcal{V}_{\text{c}}, && (4.5\text{d}) \\
\gamma_i &\ge 0, && i \in \mathcal{V}_{\text{c}}. && (4.5\text{e})
\end{aligned}
$$

Denote the solution of (4.5) as $(\bar{\mu}, \bar{\delta}, \bar{\gamma})$. Assume now that only the dual variables are known. From the dual formulation, more specifically from the constraints (4.5b), we can derive the reduced costs

$$\begin{aligned}\bar{c}^{k,r} &:= c^{kr}_{\text{tot}} - A^{kr}_1 \bar{\mu}_k - \sum_{i\in\mathcal{V}} A^{ikr}_2 \bar{\delta}_i - \sum_{i\in\mathcal{V}} A^{ikr}_3 \bar{\gamma}_i \qquad (4.6)\\ &= c^{\text{el}} \sum_{i\in\mathcal{V}} p_i + (c^{\text{w}} - \bar{\mu}_k) \sum_{i\in\mathcal{V}} \left(s_i + \sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} x_{ij} \right) - \sum_{i\in\mathcal{V}} \sum_{h\in\mathcal{H}} \left(\bar{\delta}_i m^h_{\text{L}i} + \bar{\gamma}_i m^h_{\text{U}i} \right).\end{aligned}$$

Note that $c^{kr}_{\text{tot}}, A^{kr}_1, A^{kr}_2$ and A^{kr}_3 are used in this calculation of the reduced cost. They are defined as their counterparts above, but where $p^{k,r}_i, s^{k,r}_i, x^{k,r}_{ij}$ etc. are not known anymore. Then, as we search for the lowest reduced cost, we use Equation (4.6) and formulate one of the $|\mathcal{K}|$ subproblems as

$$\begin{aligned}\min_{x,y,z,m_{\text{L}},m_{\text{U}},p,q,s,\tau} \quad & c^{\text{el}} \sum_{i\in\mathcal{V}} p_i + (c^{\text{w}} - \bar{\mu}_k) \sum_{i\in\mathcal{V}} \left(s_i + \sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} x_{ij} \right) \qquad (4.7\text{a})\\ & - \sum_{i\in\mathcal{V}} \sum_{h\in\mathcal{H}} \left(\bar{\delta}_i m^h_{\text{L}i} + \bar{\gamma}_i m^h_{\text{U}i} \right)\end{aligned}$$

subject to

1. Routing constraints,
2. Loading/unloading constraints,
3. Battery constraints,
4. Time constraints,
5. Non-negativity and binary constraints

4.2.2 Returning to the original problem

When the generation of columns has terminated, our goal is to solve a model similar to problem (4.3). We consider the set $\tilde{\mathcal{R}}_k$ from the RMP, reinstate the integer requirements on the variables $\lambda^{k,r}$ and formulate the *restricted* EFSMVRP (REFSMVRP) as

$$\min_{\lambda} \quad \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{i\in\mathcal{V}} \left(c^{\text{el}} \bar{p}^{k,r}_i + c^{\text{w}} \left(\sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} \bar{x}^{k,r}_{ij} + \bar{s}^{k,r}_i \right) \right) \lambda^{k,r} \qquad (4.7\text{ba})$$

subject to

$$\begin{aligned}\sum_{r\in\tilde{\mathcal{R}}_k} \sum_{i\in\mathcal{V}} \left(\bar{s}^{k,r}_i + \sum_{j\in\mathcal{V}:(i,j)\in\mathcal{A}} t_{ij} \bar{x}^{k,r}_{ij} \right) \lambda^{k,r} &\leq t^{\max}, && k\in\mathcal{K}, && (4.7\text{bb})\\ \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{h\in\mathcal{H}} \bar{m}^{k,h,r}_{\text{L}i} \lambda^{k,r} &\geq D_{\text{L}i}, && i\in\mathcal{V}_{\text{c}}, && (4.7\text{bc})\\ \sum_{k\in\mathcal{K}} \sum_{r\in\tilde{\mathcal{R}}_k} \sum_{h\in\mathcal{H}} \bar{m}^{k,h,r}_{\text{U}i} \lambda^{k,r} &\geq D_{\text{U}i}, && i\in\mathcal{V}_{\text{c}}, && (4.7\text{bd})\\ \lambda^{k,r} &\in \mathbb{N}, && r\in\tilde{\mathcal{R}}_k, \quad k\in\mathcal{K}. && (4.7\text{be})\end{aligned}$$

This is the optimization problem which we eventually will solve in this study. As mentioned in Section 3.3.3, its optimal value may not be equal to the optimal value of problem (4.3). Therefore, the quality of the solution will be evaluated with an upper and lower bound in Chapter 5.

4.3 Method and Stopping criteria

The commodity flow model (4.2) is always terminated with the criterion

$$\text{Stop when the optimality gap} \leq \epsilon_1 := 10^{-4}. \tag{4.7c}$$

The proposed algorithm for solving model (4.3) is shown in Figure 3.2. Termination criterion (4.7c) also applies for each optimization of a subproblem (4.7) in every iteration of the column generation algorithm.

Note that we do not need to calculate the exact route cost for the initially generated route. If we were to actually gain by using a specific route, it will be generated with its correct cost later in the column generation. In the beginning of the column generation, we include the second termination criterion

$$\text{Stop when finding one route with a negative reduced cost,} \tag{4.7d}$$

disregarding whether it is optimal or not. Criterion (4.7d) is applied until the optimal objective value of the RMP does not improve with more than $\epsilon_2 = 5\%$ between two consecutive iterations. Upon reaching that threshold, fine tuning begins and the only existing termination criterion is (4.7c). Then, for every subproblem in every iteration, we add up to five (5) routes with a negative reduced cost to $\tilde{\mathcal{R}}_k, k \in \mathcal{K}$. The column generation itself is terminated either when every subproblem exclusively yields a non-negative reduced cost or when the objective value of the RMP does not improve with more than $\epsilon_3 = 10^{-4}$ between two iterations.

4.3.1 Convergence of the column generation

The convergence rate of z^*_{MP} is important to observe in order to potentially stop column generation when new columns barely improve our solution. Let $\mathcal{R}_{\text{new},k}$ be the set of columns with negative reduced cost that are generated in a specific iteration for vehicle $k \in \mathcal{K}$. Using Equation (3.8) with a slight modification, we obtain the inequalities

$$\max\left(0, z^*_{\text{RMP}} + \sum_{k\in\mathcal{K}} \sum_{r\in\mathcal{R}_{\text{new},k}} \bar{c}^{k,r}\right) \leq z^*_{\text{MP}} \leq z^*_{\text{RMP}}, \tag{4.7e}$$

where z^*_{RMP} is the value of the solution to problem (4.4) and the reduced cost $\bar{c}^{k,r}$ is computed using the expressions in (4.6) with dual values specified for that iteration. The lower bound is bounded from below by zero as we know that $z^*_{\text{MP}} > 0$ due to a positive route cost for all routes $r \in \cup_{k\in\mathcal{K}}\mathcal{R}_k$. Note that the lower bound does not increase monotonously over the iterations [25] but is a valid lower bound which converges to z^*_{MP} upon completion of the column generation. It is important to note that when termination criterion (4.7d) is used, we may not necessarily obtain the minimum reduced cost at any subproblem optimization and consequently, the computed lower bound may be too large. However, after the removal of this criterion, the lower bound is valid.

5

Tests and results

Models (4.2), (4.4) and (4.7) and the column generation framework are all implemented in Python [29]. All optimization problems are solved using the MILP (or LP) solver provided by Gurobi [30]. All results are obtained on a PC with an Intel® Core™ i5-1145G7 2.60 GHz processor and with 16 GB of RAM. The CPU has four cores and eight logical processors with base speed 1.50 GHz. Parallelization of subproblem optimization in the column generation algorithm is implemented with the framework of the Python module Joblib.

5.1 Instance size

Instance sizes for the model (4.2) depending on the number of nodes, vehicles and trips are shown in Table 5.1. The size is measured as the number of variables, with a distinction between continuous and binary variables. As Gurobi-solvers use presolve to reduce the number of variables before optimization, the number of variables after said preprocessing is also shown in the table. For the generalized set covering model, the instance size as a function of the number of customer nodes in each subproblem is shown in Table 5.2. To be able to compare the models, it is important to observe that many subproblems are solved in multiple iterations. The number of iterations and existing routes in the set $\cup_{k\in\mathcal{K}}\tilde{\mathcal{R}}_k$ when column generation has terminated is shown in Table 5.3, indicating that the number of iterations is less than ten for every instance, meaning that an optimal solution to the model (4.4) is found relatively quickly. Furthermore, it can be seen that at most 321 routes are considered when solving (4.7b), a very small number compared to the actual number of routes that exist in an instance with twelve nodes.

Table 5.1: Instance sizes for the model (4.2) in terms of continuous and binary variables, as functions of the number of customers, vehicles and trips, before and after Gurobi's presolve is applied.

Parameters			# Variables		# Variables (after presolve)	
V_c	K	# trips	Continuous	Binary	Continuous	Binary
1	1	1	46	14	10	2
2	2	2	264	79	114	55
5	5	3	2250	786	1398	665
10	5	3	5550	2216	4167	2026
15	5	3	10350	4396	8451	4137

Table 5.2: Instance size for each subproblem (4.7) in the column generation algorithm. The instance sizes are measured in terms of continuous and binary variables as functions of the number of customers, before and after GUROBI's presolve is applied.

Parameters	Variables		Variables (after presolve)	
V_c	Continuous	Binary	Continuous	Binary
1	46	11	13	4
2	66	18	24	12
5	150	51	64	42
10	370	146	166	124
15	690	291	315	255

Table 5.3: The number of iterations performed, the number of generated routes after termination of the column generation, and the time t_{REFSMVRP} used to solve the final problem (4.7b) using these routes. Results are obtained for ten vehicles and the number of customers varying between four and ten.

V_c	# iterations	$\sum_{k\in\mathcal{K}} \lvert\mathcal{R}_k\rvert$	t_{REFSMVRP} [s]
4	6	62	0.01
5	6	114	0.01
6	5	174	0.02
7	5	173	0.03
8	6	224	0.02
9	7	242	0.02
10	9	321	0.10

5.2 Runtime performance

The time to solve instances of different sizes with the generalized set covering model with the column generation approach is shown in Figure 5.1. To ensure a feasible solution, all computations depicted in Figure 5.1 are obtained given a set of ten vehicles. Observe the logarithmic scale of the vertical axis; the computing time is exponentially growing and the instance C10-V10 is solved in almost an hour. When combining information from Figure 5.1 and Table 5.3, it appears that column generation is the most computationally expensive step of the algorithm. In fact, finding the final solution with branch-and-bound only takes a very small fraction of the total time regardless of how many columns that were generated.

At most eight vehicles were required to generate the initial routes to the RMP for the instances; see Table 5.4. We can see that larger instances tend to require more vehicles to generate a feasible initial solution with Algorithm 4.3 than what are actually needed. This affects the optimization run-time of the problem, even though the solution of the subproblems is parallelized.

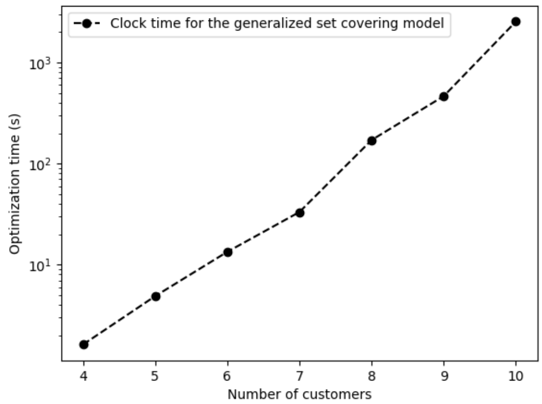

Figure 5.1: Computation time for solving the generalized set covering model using column generation as a function of the number of customers. The instances are CX-V10, $X \in \{4, 5, 6, 7, 8, 9, 10\}$, and the number of trips that each vehicle can perform is limited by the maximum working time; see constraint (4.4b).

Table 5.4: The value of the obtained solution, z^*_{REFSMVRP}, the optimal value, z^*_{RMP}, to the RMP, and the optimality gap for the instances CX-V10, for $X \in \{4, 5, 6, 7, 8, 9, 10\}$. The number of vehicles required to generate the initial, simple routes is denoted $n_{k,\text{req}}$ while $n_{k,\text{used}}$ denotes the number of vehicles used in the obtained solution.

V_c	z^*_{REFSMVRP}	z^*_{RMP}	Optimality gap [%]	$n_{k,\text{req}}$	$n_{k,\text{used}}$
10	712	690	3.2	8	5
9	624	624	0	7	5
8	597	597	0	7	4
7	486	457	6.3	6	4
6	458	429	6.8	5	4
5	351	351	0	4	4
4	275	275	0	3	3

5.2.1 Convergence of the column generation

To examine the quality of the stopping criteria for the column generation, we observe at which rate the solution to the RMP is improved. From Table 5.3 we observe that the subproblems are solved relatively few times, at most nine for the instance C10-V10, and that the time to solve (4.7b) is very short. Consequently, the optimization of subproblems to generate new routes is the time-consuming component of the algorithm. For z_{MP}, Figure 5.2 shows the upper bound z^*_{RMP} in blue and the lower bound in black, both given by the right and left inequality in (4.7e) respectively, over iterations of the column generation when solving the instance C10-V10. In the figure, the cost of every initial route is set to €1000, which is approximately ten times the real cost of the route. Note again that this cost does not need to be exact, as the same route—although with the correct cost—will be generated in case it is beneficial. A good quite tight bound is found after just two iterations, and a relatively tight upper bound is found after four iterations, just before termination criterion (4.7d) is removed for the subproblem optimization. Note that this criterion reduces the subproblem computation time significantly as we only need to find routes with a negative reduced cost, not the lowest. Consequently, the first four iterations are performed in a small fraction of the total computation time for the algorithm.

Similar tests were performed with exact costs for initial routes, but those tests did not reduce the solution time by a significant margin and yielded similar results for the bounds on z^*_{MP}. When using exact costs for the initial routes, the objective value z^*_{REFSMVRP} was often higher than when the objective value when larger costs were used. An explanation could be that, with the exact costs, the initial routes are valued too highly so that better routes are never found from column generation.

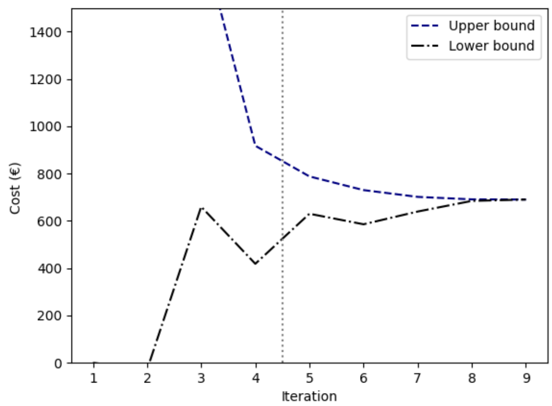

Figure 5.2: Upper and lower bound for z^*_{MP} at different iterations of the column generation algorithm when solving C10-V10. The vertical, grey line represents the removal of the termination criterion (4.7d). The bounds are given by (4.7e) but note again that when termination criterion (4.7d) is used, we do not necessarily obtain the minimum reduced cost $\bar{c}^{k,r}$ for any subproblem k. Instead, for each subproblem, the obtained reduced cost is used in (4.7e). Thus, the lower bound left of the vertical, grey line is not necessarily a valid lower bound to z^*_{MP}.

5.3 Solution quality

For the commodity flow model (4.2), it is easy to directly determine whether the obtained solution is near-optimal since the optimality gap is known; see Section 3.2.2. However, using the generalized set covering formulation, only the optimal values of the restricted problems (4.4) and (4.7b), the RMP and REFSMVRP respectively, are known. To estimate the optimal solution to (4.3), which is what we ultimately want to solve, we need to compare its objective value z^*_{EFSMVRP} to the optimal objective values z^*_{REFSMVRP} and z^*_{RMP}. We know that

$$z^*_{\mathrm{RMP}} \leq z^*_{\mathrm{EFSMVRP}} \leq z^*_{\mathrm{REFSMVRP}};$$

see (3.12) in Section 3.3.3 for the details. The obtained objective values z^*_{REFSMVRP} and z^*_{RMP} are shown in Table 5.4 together with the optimality gap computed using (3.3). For most instances, the obtained objective value z^*_{REFSMVRP} corresponds to an optimal solution to the original problem (due to a zero optimality gap). For each instance, the value of the obtained solution is at most 6.8% larger than the optimal value.

6
Discussion

The aim of our study was to formulate and solve an extension of the Electric Fleet Size and Mix Vehicle Routing Problem (EFSMVRP) with multiple trips and split pick-ups and deliveries. This extension has proven to be challenging due to the complexity of its characteristics; to account for charging, several loading possibilities, and multiple trips for a heterogeneous fleet with a fixed working time limit for each vehicle. We first proposed a commodity flow formulation, followed by a generalized set covering model with resource constraints. The latter formulation was used and solved with an algorithm including column generation and branch-and-bound.

6.1 Summary of results

By formulating the problem as a generalized set covering problem, we are able to decompose the problem and not having to consider every variable. With the proposed column generation algorithm, we are capable of solving small instances up to twelve nodes and ten vehicles in just under an hour. During the column generation, it is also shown that the optimality gap quickly becomes relatively small, but that fine tuning by changing termination criteria may further accelerate the progress. The acquired solutions to our studied problem (4.3) are regarded to be of fairly high quality and evaluated as optimal or close to optimal, due to known upper and lower bounds. A solution which is, at worst, less than 7% higher than the optimal value is obtained on all tested instances. Solutions are demonstrated to highly depend on the cargo demands but, in these instances, be almost independent of the allowed vehicle operational time, provided it is large enough so that a feasible solution exists. Generally, small vehicles tend to be used in routes with only one customer while larger vehicles visit multiple customers and become more common when cargo demands increase. Note that the results may be highly dependent of the characteristics of the instance and consequently, all results are regarded valid only for instances similar to the ones used in this study.

6.2 Discussion of method

The proposed algorithm can be divided into the following three important components:

1. Initialize routes that yield a feasible solution to problem (4.3)
2. Generate new routes to the RMP (4.4)

3. Solve model (4.7b) when all important routes have been added

In our perspective, it is important to improve the first component. The employed method has one crucial weakness: it does not always yield a feasible solution for an instance even if a solution in fact exists. Due to the characterization of the problem (time limit for vehicles; customer demands may be large compared to vehicle capacities; battery capacities), the initial routes have to be constructed carefully. A first approach can be to allow for multiple customers or charging stations in a route while still keeping track on the total time and battery level at all times. To produce routing patterns, clustering nodes based on location may be used. There are multiple studies, see e.g. [31], that uses clustering methods to solve VRPs. It could also be possible to solve a TSP for each cluster and allow for multiple trips to customers with large demands. However, none of these approaches will guarantee a feasible solution if it exists, so it is essential to explore other methods.

If we look at it from a theoretical perspective rather than a realistic one, a neat solution to the problem of finding initial routes is to use an imaginary vehicle. This vehicle may have following characteristics:

- It has a very large cargo capacity, both in terms of mass and volume,
- It can pick up or deliver cargo in every node, regardless of loading/unloading type,
- It has a large battery,
- It travels with a very high velocity and can therefore travel everywhere within the given time limit,
- The cost for using the vehicle is very high.

By using the vehicle's characteristics, we can create a route where the vehicle leaves the depot, visits all customers, fulfills their demands and returns to the depot within the permitted working time $t^{\max}$. Then, by only this route in the initial set of routes $\cup_{k\in\mathcal{K}}\tilde{\mathcal{R}}_k$ for problem (4.4), we ensure that column generation always proceed, regardless of whether the problem is feasible or not. The crucial detail is to assign a very high route cost compared to costs of all other routes, so that this artificially created route will be chosen in the final solution only if the problem instance is infeasible. This method has been implemented and tested in this study, with similar performance results but with approximately 30% longer solution times compared to the proposed method in Algorithm **??**. The cause for the slower computation time has not been investigated further, and may be a consequence of poorly designed stopping criteria for this specific method.

Moreover, further tuning of tolerances ϵ_1, ϵ_2 and ϵ_3 or other termination criteria may accelerate the algorithm, regardless of how the initial routes are created. When applying the termination criterion (4.7d) for the subproblems, only one route per vehicle is added in each iteration of the column generation algorithm. Hence, a natural improvement is to modify the criterion to stop after finding $n \in \mathbb{N} = 0, 1, \dots,$ routes with negative reduced cost, but not necessarily the n routes with the lowest reduced costs. When only termination criterion (4.7c) is active, we already do similarly, as we find the route with the lowest reduced cost and four routes that have negative reduced costs. It is reasonable, but not evident, to assume that adding slightly more routes from every subproblem optimization will accelerate the algorithm, as in our tests, the primary contribution to the total time was subproblem optimization. In our view, another useful approach could be to apply heuristic methods or other algorithms to solve the subproblems faster, at least in the beginning of the algorithm. For an overview of implemented heuristics in VRPs, see [11].

6.3 Further research

The proposed model is general and covers battery usage, multiple trips and split pick-ups and deliveries for a heterogeneous fleet of vehicles. However, it does not take time window constraints into consideration, something that in reality is reasonable and perhaps necessary to include for most customers. Time window constraints are easy to implement for the commodity flow model (4.2). However, to be able to perform column generation to solve model (4.3) with time windows included, we have to keep track of the time across multiple routes. At first glance, this is not easily fixable. For a solution approach of a model with time windows and multi-trips for a heterogeneous fleet but excluding split pick-ups and deliveries, see Seixas and Mendes [24]. Related to time windows, one can also include waiting times for loading, unloading and charging in the depot as well as at customer locations.

In this study, each subproblem (4.7) is solved as an MILP, and no further actions are taken to reduce the time of solving it. Every subproblem is an elementary shortest path problem with resource constraints (ESPPRC), an NP-hard problem [32] which is often used for subproblems when performing column generation on VRPs. Algorithms to solve the ESPPRC, both exactly and approximately, exist since many years; see [33]. For an implementation of dynamic programming to solve ESPPRCs, see e.g., [34] or [35]. In [35], Gaur and Singh solve the *cumulative* VRP, where the running cost is proportional to the vehicle weight, using column generation and a dynamic programming algorithm to solve the subproblems. Gaur and Singh observe routes defined similarly as in this study and the proposed dynamic programming algorithm solves the subproblems effectively. Therefore, the implementation of a dynamic programming algorithm could be a useful research area to search for faster subproblem optimizations.

According to Lübbecke and Desrosiers in [36], the subproblems may also be solved approximately for many iterations of the algorithm, as long as they are solved exactly in the last iteration. Essentially, we apply this idea in this study when we stop after finding just one improving route in the earlier stages of our algorithm, but it would be interesting to see whether dynamic programming could further reduce the number of iterations where the most negative reduced cost has to be found.

Additionally, we propose *partial* column generation, where the idea is to not consider every subproblem but instead a subset of them [36]. By doing so, we wish to avoid similar solutions for subproblems for which the vehicle characteristics are similar. Since a subproblem is stated for each vehicle in this study, an immediate improvement when solving large instances with many vehicles (a higher number than the possible vehicle types) would be to instead solve a subproblem for each vehicle type. For larger instances, this would reduce the number of subproblems, thus reducing the computation time. However, this study is made with very small instances (up to ten vehicles and with nine possible vehicle types excluding loading/unloading possibilities), and it is not clear how to select which subproblems should be solved. Therefore, partial column generation does not provide evident advantages computationally.

Bibliography

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management Science*, vol. 6, no. 1, pp. 80–91, 1959.

[2] M. Asghari and S. M. J. Mirzapour Al-e-hashem, "Green vehicle routing problem: A state-of-the-art review," *International Journal of Production Economics*, vol. 231, p. 107 899, 2021. DOI: https://doi.org/10.1016/j.ijpe.2020.107899.

[3] Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte, "Thirty years of heterogeneous vehicle routing," *European Journal of Operational Research*, vol. 249, no. 1, pp. 1–21, 2016. DOI: https://doi.org/10.1016/j.ejor.2015.07.020.

[4] A. Ceselli, A. Felipe, M. Ortuño, G. Righini, and G. Tirado, "A branch-and-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies," *SN Operations Research Forum*, vol. 2, no. 8, 2021. DOI: 10.1007/s43069-020-00052-x.

[5] G. Hiermann, J. Puchinger, S. Røpke, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research*, vol. 252, no. 3, pp. 995–1018, 2016. DOI: https://doi.org/10.1016/j.ejor.2016.01.038.

[6] M. Kchaou-Boujelben, "Charging station location problem: A comprehensive review on models and solution approaches," *Transportation Research Part C: Emerging Technologies*, vol. 132, p. 103 376, 2021. DOI: https://doi.org/10.1016/j.trc.2021.103376.

[7] T. Ghandriz, B. Jacobson, M. Islam, J. Hellgren, and L. Laine, "Transportation-mission-based optimization of heterogeneous heavy-vehicle fleet including electrified propulsion," *Energies*, vol. 14, no. 11, 2021. DOI: 10.3390/en14113221.

[8] European Union. "Employing road transport workers: Driving time and rest periods," Accessed: Jul. 15, 2024.: https://europa.eu/youreurope/business/human-resources/transport-sector-workers/road-transportation-workers/index_en.htm.

[9] E. Choi and D.-W. Tcha, "A column generation approach to the heterogeneous fleet vehicle routing problem," *Computers & Operations Research*, vol. 34, pp. 2080–2095, 2007. DOI: 10.1016/j.cor.2005.08.002.

[10] J. K. Lenstra and A. H. G. Rinnooy Kan, "Complexity of vehicle routing and scheduling problems," *Networks*, vol. 11, pp. 221–227, 1981.

[11] B. Golden, X. Wang, and E. Wasil, *The Evolution of the Vehicle Routing Problem: A Survey of VRP Research and Practice from 2005 to 2022*. Springer Nature Switzerland, 2023. DOI: 10.1007/978-3-031-18716-2_1.

[12] R. Tavakkoli-Moghaddam, N. Safaei, M. M. O. Kah, and M. Rabbani, "A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing," *Journal of the Franklin Institute*, vol. 344, no. 5, pp. 406–425, 2007, SI: Modeling, Simulation and Applied Optimization Part II. DOI: `https://doi.org/10.1016/j.jfranklin.2005.12.002`.

[13] D. Cattaruzza, N. Absi, and D. Feillet, "Vehicle routing problems with multiple trips," *4OR*, vol. 14, no. 3, pp. 223–259, 2016. DOI: `10.1007/s10288-016-0306-2`.

[14] B. Kallehauge, J. Larsen, O. B. Madsen, and M. M. Solomon, "Vehicle routing problem with time windows," in *Column Generation*, G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds. Boston, MA: Springer US, 2005, pp. 67–98. DOI: `10.1007/0-387-25486-2_3`.

[15] N. Huang, H. Qin, G. Xu, and F. Wan, "An enhanced exact algorithm for the multi-trip vehicle routing problem with time windows and capacitated unloading station," *Computers & Operations Research*, vol. 168, p. 106 688, 2024. DOI: `https://doi.org/10.1016/j.cor.2024.106688`.

[16] Y. Mo, K. Yang, S. Han, and S. Gupta, "Multi-period heterogeneous fleet vehicle routing problem with self-pickup point selection: A last-mile delivery scenario in urban and rural areas," *Annals of Operations Research*, pp. 1–35, May 2024. DOI: `10.1007/s10479-024-06011-7`.

[17] I. Kucukoglu, R. Dewil, and D. Cattrysse, "The electric vehicle routing problem and its variations: A literature review," *Computers Industrial Engineering*, vol. 161, p. 107 650, 2021. DOI: `https://doi.org/10.1016/j.cie.2021.107650`.

[18] A. N. Letchford and J.-J. Salazar-González, "Stronger multi-commodity flow formulations of the capacitated vehicle routing problem," *European Journal of Operational Research*, vol. 244, no. 3, pp. 730–738, 2015. DOI: `https://doi.org/10.1016/j.ejor.2015.02.028`.

[19] F. Arnold, M. Gendreau, and K. Sörensen, "Efficiently solving very large-scale routing problems," *Computers & Operations Research*, vol. 107, pp. 32–42, 2019. DOI: `https://doi.org/10.1016/j.cor.2019.03.006`.

[20] G. D. Konstantakopoulos, S. Gayialis, and E. Kechagias, "Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification," *Operational Research*, pp. 1–30, 2022. DOI: `10.1007/s12351-020-00600-7`.

[21] T. Ghandriz, B. Jacobson, L. Laine, and J. Hellgren, "Impact of automated driving systems on road freight transport and electrified propulsion of heavy vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 115, 2020. DOI: `https://doi.org/10.1016/j.trc.2020.102610`.

[22] M. Padberg, *Linear Optimization and Extensions*. Springer Berlin Heidelberg, 1999. DOI: `10.1007/978-3-662-12273-0_6`.

[23] K. Hoffman and M. Padberg, "Set covering, packing and partitioning problemsset covering, packing and partitioning problems," in *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, Eds. Springer US, 2001. DOI: `10.1007/0-306-48332-7_459`.

[24] M. Seixas and A. Mendes, "Column generation for a multitrip vehicle routing problem with time windows, driver work hours, and heterogeneous fleet," *Mathematical Problems in Engineering*, vol. 2013, 2013. DOI: 10.1155/2013/824961.

[25] J. Desrosiers and M. E. Lübbecke, "A primer in column generation," in *Column Generation*, G. Desaulniers, J. Desrosiers, and M. M. Solomon, Eds. Boston, MA: Springer US, 2005, pp. 1–32. DOI: 10.1007/0-387-25486-2_1.

[26] J. Desrosiers, M. Lübbecke, G. Desaulniers, and J.-B. Gauthier, "Branch-and-price," Groupe d'études et de recherche en analyse des décisions, Les Cahiers du GERAD G-2024-36, 2024.: https://www.gerad.ca/en/papers/G-2024-36, published.

[27] J. Kindstrand and L. Nordgren, *Battery dimensioning for hybrid vehicles in a routing application*, https://odr.chalmers.se/items/64f4208b-1844-4cda-8ee1-b1b08b773a5e, M.Sc. thesis, Chalmers University of Technology, 2018.

[28] O. Arslan and O. E. Karaşan, "A Benders decomposition approach for the charging station location problem with plug-in hybrid electric vehicles," *Transportation Research Part B: Methodological*, vol. 93, pp. 670–695, 2016. DOI: https://doi.org/10.1016/j.trb.2016.09.001.

[29] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. CreateSpace, 2009.: https://docs.python.org/3/reference/index.html.

[30] Gurobi Optimization, LLC, *Gurobi Optimizer Reference Manual*, 2024.: https://www.gurobi.com/documentation/current/refman/index.html.

[31] R. Dondo and J. Cerdá, "A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows," *European Journal of Operational Research*, vol. 176, no. 3, pp. 1478–1507, 2007. DOI: https://doi.org/10.1016/j.ejor.2004.07.077.

[32] M. Desrochers, J. Desrosiers, and M. Solomon, "A new optimization algorithm for the vehicle routing problem with time windows," *Operations Research*, vol. 40, no. 2, pp. 342–354, 1992.

[33] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, "An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems," *Networks*, vol. 44, pp. 216–229, Oct. 2004.

[34] S. Eriksson Barman, P. Lindroth, and A.-B. Strömberg, "Modeling and solving vehicle routing problems with many available vehicle types," in *Optimization, Control, and Applications in the Information Age*, A. Migdalas and A. Karakitsiou, Eds., Cham: Springer International Publishing, 2015, pp. 113–138.

[35] D. R. Gaur and R. R. Singh, "Cumulative vehicle routing problem: A column generation approach," in *Algorithms and Discrete Applied Mathematics*, S. Ganguly and R. Krishnamurti, Eds., Cham: Springer International Publishing, 2015, pp. 262–274.

[36] M. E. Lübbecke and J. Desrosiers, "Selected topics in column generation," *Operations Research*, vol. 53, pp. 1007–1023, 2005. DOI: 10.1287/opre.1050.0234.