Chair of Applied Mathematics

Master's Thesis

A Decomposition Method for the Deterministic Flow Refueling Location Problem (DFRLP)

Marcel Höller, BSc

May 2024

MONTANUNIVERSITÄT LEOBEN
www.unileoben.ac.at

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich diese Arbeit selbstständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt, den Einsatz von generativen Methoden und Modellen der künstlichen Intelligenz vollständig und wahrheitsgetreu ausgewiesen habe, und mich auch sonst keiner unerlaubten Hilfsmittel bedient habe.

Ich erkläre, dass ich den Satzungsteil „Gute wissenschaftliche Praxis" der Montanuniversität Leoben gelesen, verstanden und befolgt habe.

Weiters erkläre ich, dass die elektronische und gedruckte Version der eingereichten wissenschaftlichen Abschlussarbeit formal und inhaltlich identisch sind.

Datum 14.05.2024

Unterschrift Verfasser/in
Marcel Höller

Kurzfassung

Der Verkehrssektor ist ein wesentlicher Treiber des Klimawandels und zeichnet für einen beträchtlichen Anteil der weltweiten CO_2-Emissionen verantwortlich. Batterieelektrische Fahrzeuge bieten eine vielversprechende Lösung zur Emissionsreduktion, erfordern jedoch für eine breite Akzeptanz ein gut ausgebautes Ladenetz. Das Deterministic Flow Refueling Location Problem (DFRLP) befasst sich mit der Optimierung der Platzierung von Ladestationen unter Berücksichtigung der Verkehrsströme, um die Abdeckung des Ladebedarfs zu maximieren.

Diese Arbeit adressiert die Kombination zweier bestehender Erweiterungen des DFRLP. Diese berücksichtigen die Dimensionierung von Ladestationen mit begrenzter Kapazität, sowie die Kostenheterogenität in städtischen, vorstädtischen und ländlichen Gebieten. Zur effizienten Lösung dieses erweiterten DFRLP wird eine problemspezifische Dekompositionsmethode entwickelt und angewendet.

Die entwickelte Dekompositionsmethode zerlegt einen gegebenen Graphen durch das Entfernen der Kanten mit dem kleinsten Verkehrsvolumen, bis der Graph in kleinere Cluster zerfällt, auf die das erweiterte DFRLP effizient angewendet werden kann. Die Wirksamkeit der Dekompositionsmethode wird durch umfangreiche numerische Experimente demonstriert. Die Ergebnisse zeigen, dass die Lösungsqualität bei einer deutlichen Laufzeitreduktion annähernd der Optimallösung der vollen Datensätze entspricht.

Die vorliegende Arbeit leistet einen Beitrag zur Optimierung der Ladeinfrastruktur für Elektrofahrzeuge und bietet ein praktisches Werkzeug für die Entscheidungsfindung im Bereich der Verkehrsplanung. Die vorgeschlagene Methode kann Entscheidungsträger, Infrastrukturplaner und private Investoren dabei unterstützen, die Platzierung und Dimensionierung von Ladestationen zu optimieren, um eine nachhaltige Verkehrszukunft zu ermöglichen.

Abstract

The transport sector is a significant driver of climate change and is responsible for a substantial proportion of global CO_2 emissions. Battery electric vehicles offer a promising solution for reducing emissions, but require a well-developed charging network for widespread acceptance. The Deterministic Flow Refueling Location Problem (DFRLP) deals with optimizing the placement of charging stations considering traffic flows in order to maximize the coverage of charging demand.

This thesis addresses the combination of two existing extensions of the DFRLP. These consider the sizing of charging stations with limited capacity, as well as the cost heterogeneity in urban, suburban and rural areas. A problem-specific decomposition method is developed and applied to efficiently solve this extended DFRLP.

The developed decomposition method decomposes a given graph by removing the edges with the smallest traffic volume until the graph is decomposed into smaller clusters to which the extended DFRLP can be efficiently applied. The effectiveness of the decomposition method is demonstrated through extensive numerical experiments. The results show that the solution quality is close to the optimal solution of the full data sets with a significant reduction in runtime.

This work contributes to the optimization of electric vehicle charging infrastructure and provides a practical tool for decision making in the field of transportation planning. The proposed method can assist decision makers, infrastructure planners and private investors in optimizing the placement and sizing of charging stations to enable a sustainable transportation future.

Table of Contents

Kurzfassung **I**

Abstract **II**

List of Figures **V**

List of Tables **VI**

List of Abbreviations **VII**

1 Introduction **1**

1.1 Related Literature . . . 2

1.2 Methodology . . . 3

2 ILP models **5**

2.1 Formal problem definition of the Deterministic Flow Refueling Location Problem (DFRLP) . . . 5

2.2 Minimum Flow Volume Coverage (MC-DFRLP) . . . 9

2.3 Location-dependent costs per charging station (LC-DFRLP) . . . 10

2.4 Capacitated DFRLP (C-DFRLP) . . . 11

2.5 Capacitated Minimum Flow Volume Coverage DFRLP (C+MC-DFRLP) . 14

2.6 Combination of C+MC-DFRLP and LC-DFRLP (C+LC-DFRLP) . . . 15

2.6.1 Step 1: Maximize Covered Flow Volume . . . 15

2.6.2 Step 2: Minimize Cost . . . 17

2.6.3 Scalability and Computational Efficiency Analysis . . . 18

2.6.4 LP-based solution attempts . . . 19

2.6.4.1 Quadratic range constraint . . . 19

2.6.4.2 Alternative linearization of the range constraint . . . 20

3 Decomposition method **21**

3.1 Development Process and Initial Approaches . . . 21

3.2 The proposed decomposition method . . . 23

4 Test Setup **26**

4.1 Software and Hardware 26
4.2 Benchmark Instances 26
4.2.1 Determination of the charging pole capacity 27
4.2.2 Classification of the location types 28
4.3 Parameters and Evaluation Layout 29
4.3.1 Cost parameters 29
4.3.2 General Model Parameters 30
4.3.3 Clustering-Specific Parameter 30
4.4 Challenges, Assumptions and Limitations 32
4.5 Evaluation Layout 33

5 Numerical Results 35
5.1 Optimal solutions for the full dataset 36
5.2 Solution Quality Comparison 36
5.2.1 Maximize achievable CFV 36
5.2.2 Minimize Cost while maintaining maximum CFV 37
5.2.3 Maximize Covered Flow Volume (CFV) with Incremental Budgets . 37
5.3 Runtime Comparison 39
5.3.1 Maximize achievable CFV 39
5.3.2 Minimize Cost while maintaining maximum CFV 40
5.3.3 Maximize CFV with Incremental Budgets 41

6 Conclusions and Future Research 43

Literature XII

List of Figures

1 The Different Types of Cycle Segments, Adopted From de Vries and Duijzer (2017, p. 104) 6

2 Logarithmic Plot of the Solve Time for the Datasets 18

3 Flowchart of the Decomposition Approach 23

4 The Classification of Locations Into Different Categories, Redrawn From Kastner et al. (2023, p. 20) 28

5 Total Runtime Dependent on the Maximum Cluster Size 31

6 Max. Coverage and Min. Cost Dependent on the Maximum Cluster Size . 32

List of Tables

1 Solve Time of Both Steps of the Model on the Full Dataset 18

2 Characteristics of the Benchmark Instances (Staněk et al. 2023) 27
3 Charging Pole Capacity per Dataset (Staněk et al. 2023) 28
4 Location Type Parameters (Kastner et al. 2023, p. 21) 29
5 Costs by Location Type 29
6 General Model Parameters 30
7 Decomposition Parameters 30

8 Optimal Objective Values and Solve Times for Full Dataset Solutions . . . 36
9 Objective Values for Step 1 of the C+LC-DFRLP 37
10 Objective Values for Step 2 of the C+LC-DFRLP 37
11 Objective Values for Step 1 of the C+LC-DFRLP With Incremental Budget Values 38
12 Summary of Differences in Objective Function by Budget Level 39
13 Runtime for Step 1 of the C+LC-DFRLP: Full Dataset vs. Decomposition Method 40
14 Runtime for Step 2 of the C+LC-DFRLP: Full Dataset vs. Decomposition Method 40
15 Solve times for Step 1 of the Capacitated Location-dependent costs per charging station (C+LC-DFRLP) With Incremental Budget Values 41
16 Summary of Solve Time Differences by Budget Level 42

17 Declaration of the Usage of AI-based Tools XV

List of Abbreviations

AMPL A Mathematical Programming Language

API Application Programming Interface

BEV Battery Electric Vehicle

C-DFRLP Capacitated DFRLP

C+LC-DFRLP Capacitated Location-dependent costs per charging station

C+MC-DFRLP Capacitated Minimum Flow Volume Coverage DFRLP

CFV Covered Flow Volume

DFRLP Deterministic Flow Refueling Location Problem

DFS Depth-First Search

FCLP Flow Capturing Location Problem

FRLP Flow Refueling Location Problem

GHG Greenhouse Gas

ILP Integer Linear Program

LC-DFRLP Location-dependent costs per charging station

LP Linear Program

MC-DFRLP Minimum Flow Volume Coverage

MILP Mixed Integer Linear Program

OD Origin/Destination

TFV Total Flow Volume

1 Introduction

Climate change poses a severe and undeniable threat to the planet, driven primarily by human activities that result in the emission of Greenhouse Gases (GHGs). The scientific consensus strongly supports the notion that GHG emissions, particularly CO_2, are the primary drivers of climate change (cf. Lee et al. 2023, p. 4). These emissions are causing global temperatures to rise, leading to a cascade of negative consequences such as rising sea levels, more extreme weather events and disruption of ecosystems. The transportation sector is a significant contributor to climate change, accounting for approximately 15% of all GHG emissions worldwide in 2019, or approximately 8.7 billion tons of CO_2-equivalent annually (cf. Lee et al. 2023, p. 44). Transitioning to cleaner transportation solutions is a critical step in the mitigation of climate change.

Battery Electric Vehicles (BEVs), when powered by electricity generated from low GHG emission sources, hold significant promise for reducing GHG emissions from land-based transport on a life cycle basis. Recent advancements in battery technology are enhancing the feasibility of electrifying heavy-duty trucks, which can complement existing electric rail systems and further reduce emissions across the transportation sector (cf. Lee et al. 2023, p. 29).

However, widespread adoption of BEVs still relies heavily on the development and expansion of a robust charging network. While the costs associated with BEVs are on a downward trend and their adoption is accelerating, continued investments in the infrastructure are required to increase the scale and speed of deployment (cf. Lee et al. 2023, p. 105). Without a well-planned network of charging stations, range anxiety remains a psychological factor that hinders the adoption of BEVs. To address this, the locations of charging stations have to be planned strategically, in order to provide the best possible coverage of the charging demand. The Deterministic Flow Refueling Location Problem (DFRLP) is an optimization problem that can be used to achieve this goal. It involves determining the optimal placement of charging stations to maximize the coverage of traffic flows within a given network. This thesis builds on the DFRLP by combining two existing extensions: the C-DFRLP, which incorporates capacity constraints, and the LC-DFRLP, which considers location-dependent costs. The combined model, referred to as C+LC-DFRLP, offers a more realistic representation of real-world challenges associated with BEV infrastructure planning.

The proposed C+LC-DFRLP incorporates both fixed and variable costs associated with

establishing charging stations of different sizes in urban, suburban and rural areas. Fixed costs include expenses such as land acquisition and the establishment of an electricity supply, whereas variable costs account for the installation of additional charging poles. By considering these factors, this model extension aims to achieve maximum coverage at minimal cost.
To address the computational challenges posed by the C+LC-DFRLP, a problem-specific decomposition method is developed. This method involves breaking down the problem into smaller subproblems that can be solved more efficiently. The effectiveness and performance of this decomposition approach is demonstrated through extensive numerical experiments, which show that it significantly reduces the solve time while maintaining solution quality close to the optimum for the full dataset.

This thesis is structured as follows: Chapter 2, *ILP Models*, introduces the DFRLP, its relevant extensions and the C+LC-DFRLP. It also analyzes the scalability of the model, pointing out the necessity to improve the runtime behavior and initial improvement attempts. Chapter 3, *Decomposition Method*, outlines the developed problem-specific decomposition method and its development process. Chapter 4, *Test Setup*, describes the experimental setup, including software and hardware details, presents the utilized benchmark instances, and lists the specific parameters used for the evaluations. Chapter 5, *Numerical Results*, provides the results of the experiments, where the performance of the decomposition approach is compared to the performance of the model on the full dataset. Finally, Chapter 6, *Conclusions and Future Research*, summarizes this thesis and suggests areas of future improvement.

1.1 Related Literature

In the field of optimization models for BEVs and other alternative fuel vehicles, there is a substantial body of research investigating the optimal placement of charging stations. Kchaou-Boujelben (2021) performed a comprehensive literature review of the existing models in this space, classified by node-based and flow-based approaches, and provided an overview of utilized methods to address the computational difficulties of the models. The two main approaches, node-based and flow-based, differ in their methodology for allocating the recharging demand. In node-based models, the recharging demand is assigned to the nodes of a network. In contrast, in flow-based models, demand is assigned to flows between Origin/Destination (OD) nodes (cf. Kchaou-Boujelben 2021, p. 4).
The first flow-based models were introduced by Hodgson (1990) with the formulation of the Flow Capturing Location Problem (FCLP), and significant developments have taken place since. In the FCLP, a single facility along the path is considered to *cover* that flow (cf. Hodgson 1990, p. 271). However, this model does not consider the limited driv-

ing range of BEVs, which may require multiple charging stops. Kuby and Lim (2005) addressed this issue by presenting the Flow Refueling Location Problem (FRLP), which considers driving range limitations.

This first FRLP model was proposed as a two-step approach: in the first step, all possible combinations of nodes that can refuel a path are generated. These combinations are then used in the subsequent step to solve a Mixed Integer Linear Program (MILP) to select the optimal locations from the pre-generated combinations (cf. Kuby and Lim 2005, pp. 132–134). Due to this pre-generation, this model is computationally extensive and becomes impractical for larger problem instances. To overcome this limitation, Lim and Kuby (2010) proposed heuristic algorithms to solve the FRLP (cf. Lim and Kuby 2010, p. 51).

Capar and Kuby (2012) introduced a MILP formulation of the FRLP that no longer requires pre-generated location combinations. The logic behind the pre-generation was incorporated directly into the model's constraints. However, the model formulation does not explicitly consider driving range (cf. Capar and Kuby 2012, pp. 625–626).

Upchurch et al. (2009) addressed the limitations of previous models by introducing capacity constraints, limiting the number of BEVs that can recharge at a single charging station. Another, more accurate extension considering the limited capacity, was proposed by Hosseini and MirHassani (2017). In their model, capacity is defined by considering the amount of "fuel" consumed at each station. In their capacitated formulation of the FRLP, Wang and Lin (2013) introduced budget constraints and different types of charging stations.

The Deterministic Flow Refueling Location Problem (DFRLP) was presented by de Vries and Duijzer (2017), and explicitly incorporates the driving range of BEVs as input parameter. In their study, Staněk et al. (2023) introduce several extensions to the DFRLP model, which set the stage for this Master's Thesis.

1.2 Methodology

This thesis introduces an extension to the DFRLP, the C+LC-DFRLP, which is described in detail in Section 2.6, and is a combination of existing extensions. The objective of this thesis is to explore solutions to the scalability limitations of this extended model. To efficiently solve the model, a straightforward problem-specific decomposition approach was developed. This decomposition method breaks down the problem by iteratively removing flow connections of the graph until a certain size threshold is met, thereby decomposing the graph into smaller subgraphs. The model is then solved on these small subgraphs, with the solutions from these subgraphs being reused to accelerate the solve process for the full graph. The models are modeled using AMPL, and a Java project is used to implement the decomposition method. To assess the efficacy of the decomposition approach in terms of both runtime and solution quality, several benchmarks are utilized. The C+LC-DFRLP is applied to the full benchmark datasets with and without using the decomposition method, enabling performance comparison. Section 4.5 details which tests were performed.

2 ILP models

This thesis is built on the work of Staněk et al. (2023), whose research is based on the DFRLP introduced by de Vries and Duijzer (2017). The DFRLP is an Integer Linear Program (ILP) model for the optimization of charging station locations for BEVs, and will be described in detail in Section 2.1. The basic DFRLP model is designed to maximize the covered traffic flow volume, given a fixed number of charging facilities[1] to locate. The later introduced model extensions use the same model basis, with differing modifications, and will be detailed in the remainder of this chapter.

2.1 Formal problem definition of the Deterministic Flow Refueling Location Problem (DFRLP)

This problem was originally defined by de Vries and Duijzer (2017), who described it in the context of a graph $G(L, E)$, where L is the set of nodes (locations) and E is the set of edges between these nodes. The locations L are a union of the three subsets O (origins), K (potential facility locations), and D (destinations).

The traffic in the graph is depicted as cyclic flows F. Every flow $f \in F$ is defined by its origin $O_f \in O$, its flow volume $v_f \in \mathbb{N}$ (i.e., the number of drivers of this flow), its destination $D_f \in D$ and the desired path, consisting of a set of potential facility locations $K_f \in K$ that are passed on their way. This definition results in all drivers of a flow using the same path to get from their origin to their destination and back, cyclically, driving by the same potential facility locations and without deviating from the route (cf. de Vries and Duijzer 2017, p. 103).

The driving range of the vehicles is denoted by the parameter R, which is the same for all the vehicles in the network. Recharging can only take place at opened charging facilities along the flow. A flow is considered *covered*, if the vehicles of this flow can drive from their origin to their destination and back, without running out of battery, which is only the case if the vehicle's driving range is greater than the distance between two consecutive charging facilities (cf. de Vries and Duijzer 2017, p. 103).

[1] For the remainder of this thesis, charging facilities and charging stations are terms that can be used interchangeably.

Staněk et al. (2023) slightly modify the original definition for cycle segments:

Definition 1. *A cycle segment of the flow f is identified by two nodes k and l and has corresponding distances t_{kl} as defined below:*

- *If $k = O_f$ and $l \in K_f$, the cycle segment defined by these two nodes is the path $l \to O_f \to l$ and its distance t_{kl} is given by the distance from l via O_f to l, both along f.*
- *For all $k, l \in K_f$, where k occurs before l in the flow f on the way from O_f to D_f, the cycle segment defined by these two nodes is the path $k \to l$ and its distance t_{kl} is given by the distance from k to l, both along f.*
- *If $k \in K_f$ and $l = D_f$, the cycle segment defined by these two nodes is the path $k \to D_f \to k$ and its distance t_{kl} is given by the distance from k via D_f to k, both along f.*
- *In the case where $k = O_f$ and $l = D_f$, no cycle segment is defined and $t_{kl} = \infty$.*

Figure 1 provides a visual representation of the various cycle segments:

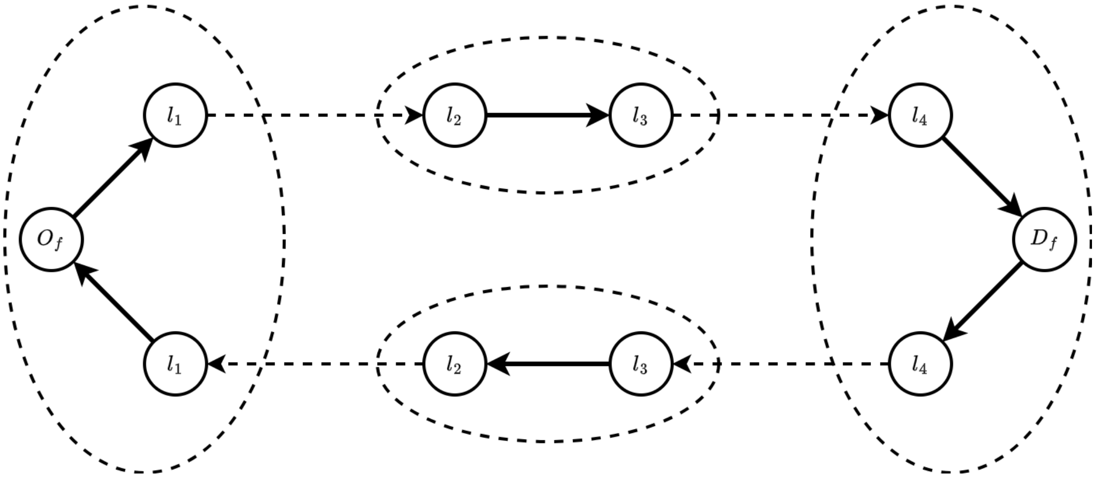

Figure 1: The Different Types of Cycle Segments, Adopted From de Vries and Duijzer (2017, p. 104)

The illustration above outlines the three distinct types of cycle segments, each cycle segment being encircled by a dashed line. Following Example 1 provides a concrete example that utilizes the depicted cycle segments:

Example 1. *Consider a flow f from O_f to D_f. The flow follows the path $O_f \to l_1 \to l_2 \to l_3 \to l_4 \to D_f$. The set K_f consists of nodes l_1, l_2, l_3 and l_4.*

- *For $k = O_f$ and $l = l_1$, the cycle segment is $l_1 \to O_f \to l_1$. If the distance from $O_f \to l_1$ is 1, the distance of the cycle segment $t_{O_f,l_1} = 1 + 1 = 2$.*

- *For $k = l_2$ and $l = l_3$, the cycle segment is $l_2 \rightarrow l_3$. If the distance from $l_2 \rightarrow l_3$ is 4, then $t_{l_2,l_3} = 4$.*
- *For $k = l_4$ and $l = D_f$, the cycle segment is $l_4 \rightarrow D_f \rightarrow l_4$. If the distance from $l_4 \rightarrow D_f$ is 1, then $t_{l_4,D_f} = 1 + 1 = 2$.*
- *For $k = O_f$ and $l = D_f$, no cycle segment is defined, and $t_{O_f,D_f} = \infty$.*

In the DFRLP, charging stations cannot be located at OD nodes due to the model's formulation, resulting in doubled distances for all cycle segments that involve an OD node and a potential facility location, as described before. To enable charging stations at OD nodes, it is necessary to replace the original O_f and/or D_f with a dummy facility location. This dummy location is then connected to the original O_f and/or D_f by an edge with zero distance (cf. Staněk et al. 2023).

The following parameters and decision variables are consistent with those used by Staněk et al. (2023) and de Vries and Duijzer (2017, p. 104):

Parameters:

• F	Set of flows
• O ($O_f \in O$)	Set of origins (origin of flow f)
• K ($K_f \in K$)	Set of potential facility locations (along flow f)
• D ($D_f \in D$)	Set of destinations (destination of flow f)
• $v_f \in \mathbb{N}$	Volume of flow f
• L ($L_f \in L$)	Set of locations, i.e., $L = O \cup K \cup D$ (set of locations along flow f, i.e., $L_f = \{O_f\} \cup K_f \cup \{D_f\}$)
• E	Set of edges between locations
• $L^-_{kf} \subsetneq L$ ($L^+_{kf} \subsetneq L$)	Set of locations along flow f passed before (after) location k on a trip from O_f to D_f
• $p \in \mathbb{N}$	Number of new facilities to locate
• $R \in \mathbb{N}$	Driving range
• $t_{kl} \geq 0$	Length of the cycle segment identified by locations k and l
• M	A sufficiently large constant used for linearizing the driving range constraint

Decision Variables:

- $x_k \in \{0,1\}$ 1 if a facility is placed at location k and 0 otherwise
- $y_f \in \{0,1\}$ 1 if flow f is covered and 0 otherwise
- $i_{klf} \in \{0,1\}$ 1 if cycle segment k, l is used in flow f and 0 otherwise

The original model formulation has been slightly adapted by Staněk et al. (2023):

Model:

$$\max \sum_{f \in F} v_f y_f \tag{2.1}$$

$$\text{s.t.} \quad \sum_{k \in K} x_k = p \tag{2.2}$$

$$\sum_{l \in L^+_{kf}} i_{klf} t_{kl} - (1 - y_f)M \le R \qquad f \in F,\ k \in \{O_f \cup K_f\} \tag{2.3}$$

$$\sum_{l \in L^+_{kf}} i_{klf} = x_k \qquad f \in F,\ k \in K_f \tag{2.4}$$

$$\sum_{l \in L^+_{O_f f}} i_{O_f l f} = 1 \qquad f \in F \tag{2.5}$$

$$\sum_{k \in L^-_{lf}} i_{klf} = x_l \qquad f \in F,\ l \in K_f \tag{2.6}$$

$$\sum_{k \in L^-_{D_f f}} i_{k D_f f} = 1 \qquad f \in F \tag{2.7}$$

$$i_{klf} \in \{0,1\} \quad f \in F,\ k \in \{O_f \cup K_f\},\ l \in L^+_{kf} \tag{2.8}$$

$$x_k,\ y_f \in \{0,1\} \quad f \in F,\ k \in K \tag{2.9}$$

The objective function (2.1) aims to maximize the total flow volume covered. Constraint (2.2) defines the number of charging facilities that should be built. Constraint (2.3) ensures that the length of the used segments is less than or equal to the range of the vehicle. Constraints (2.4) and (2.6) ensure that a cycle segment is only used if it is connected to an opened charging facility. Constraints (2.5) and (2.7) ensure that for each flow, only one segment can be used to get to the origin or destination of the flow. The decision variables are defined in (2.8) and (2.9).

2.2 Minimum Flow Volume Coverage (MC-DFRLP)

Instead of trying to maximize the covered flow volume with a fixed set of opened charging stations, this model's approach is to efficiently allocate a minimum number of opened charging stations while maintaining a certain coverage level. The coverage level, denoted as C, represents the minimum proportion of drivers who can complete their trips without running out of fuel, ensuring a reliable and efficient charging network (cf. Kastner et al. 2023, p. 8).[2]

Additional parameter:

- $C \in [0, 1]$	Minimum coverage level as proportion of the total flow volume

Model:

$$\min \quad \sum_{k \in K} x_k \tag{2.10}$$

$$\text{s.t.} \quad \frac{\sum_{f \in F} v_f y_f}{\sum_{f \in F} v_f} \geq C \tag{2.11}$$

$$(2.3) - (2.9).$$

The objective function (2.10) minimizes the number of opened stations. The constraint (2.11) enforces that the proportion of covered flow volume is greater than or equal to the parameter C. The remaining constraints (2.3) − (2.9) stay the same as in the DFRLP.

[2] This model is not part of the paper by Staněk et al. (2023), and was published in the technical report by Kastner et al. (2023).

2.3 Location-dependent costs per charging station (LC-DFRLP)

This model extension, introduced by Staněk et al. (2023), enhances the basic DFRLP by considering different one-off costs for the construction of charging stations, depending on the location. This is based on the idea that the land costs in urban, suburban and rural environments usually differ. This model extension uses location-dependent construction costs as an additional parameter for the optimization model (cf. Staněk et al. 2023).

Additional parameters:

- $c_k \geq 0$ Construction costs per charging station at location k
- $B > 0$ Available total budget

Model:

$$\max \quad \sum_{f \in F} v_f y_f \tag{2.1}$$

$$\text{s.t.} \quad \sum_{k \in K} c_k x_k \leq B \tag{2.12}$$

$$(2.3) - (2.9).$$

The objective function (2.1) remains the same as in the DFRLP, which tries to maximize the covered flow volume. The total cost of construction for opened charging stations is limited to the budget parameter B by constraint (2.12). The constraints (2.3) – (2.9) remain the same as in the DFRLP.

2.4 Capacitated DFRLP (C-DFRLP)

In this extension, Staněk et al. (2023) take into account that the capacity of each charging facility is limited. A charging facility consists of one or more charging poles, and the capacity of each charging facility depends on the number of available charging poles at the corresponding facility. It is assumed that each charging pole has a limited capacity. Therefore, more charging poles are needed at charging stations with higher demand, and fewer charging poles at less frequented charging stations, to avoid an excess of idle charging poles (cf. Staněk et al. 2023).

Based on the ideas of Upchurch et al. (2009) and Hosseini and MirHassani (2017), the amount of required energy at each charging station along the flow of a vehicle is dependent on the current battery level. Energy consumption at each station is assumed to be linearly proportional to the distance travelled since the last refueling process. The capacity of each pole is defined as energy output in terms of total driving range per pole and period. It is assumed that batteries are always filled to full capacity. Another assumption is that flows are divisible, meaning that the coverage of flows may be lower than 100%. For this, the decision variable z_f is introduced in opposition to the binary variable y_f (cf. Staněk et al. 2023).

Additional decision variables:

- $n_k \in \mathbb{N}_0$ Number of charging poles at location k
- $z_f \in [0,1]$ Proportion of flow f that is covered
- $w_{klf} \in [0,1]$ Auxiliary variable for linearization

Additional parameters:

- $Cap \in \mathbb{N}$ Capacity of charging pole given as the amount of available energy in distance units
- $e_f \in (0,1]$ Positive range-based refueling frequency ≤ 1 to cover flow f per observation period for flows with $2\tilde{t}_{O_f D_f} \leq R$; otherwise, $e_f = 1$. Specifically,

$$e_f = \frac{1}{\max\left\{1, \left\lfloor \frac{R}{2 \cdot \tilde{t}_{O_f D_f}} \right\rfloor\right\}}$$

where $\tilde{t}_{O_f D_f} > 0$ represents the real distance between origin and destination (note that $\tilde{t}_{O_f D_f} \neq t_{O_f D_f}$, according to the definition $t_{O_f D_f} = \infty$ in Definition 1). This parameter is needed to model short trips properly.

- $M_k \in \mathbb{N}$ Location-dependent maximum number of charging poles at a charging station (if $M_k = M$ for all $k \in K$, use M)
- $S \in \mathbb{N}$ Total number of charging poles to locate

Model:

$$\max \sum_{f \in F} v_f z_f \tag{2.13}$$

$$\text{s.t.} \quad \sum_{k \in K} n_k = S \tag{2.14}$$

$$x_k \leq n_k \qquad k \in K \tag{2.15}$$

$$n_k \leq M_k x_k \qquad k \in K \tag{2.16}$$

$$\sum_{f \in F\colon\, k \in K_f} \left(\sum_{l \in L^-_{kf}} t_{lk} v_f e_f w_{lkf} + \sum_{l \in L^+_{kf}} t_{kl} v_f e_f w_{klf} \right) \leq Cap \cdot n_k \qquad k \in K \tag{2.17}$$

$$w_{klf} \leq i_{klf} \qquad f \in F, k \in K_f, l \in L^+_{kf} \tag{2.18}$$

$$w_{klf} \leq z_f \qquad f \in F, k \in K_f, l \in L^+_{kf} \tag{2.19}$$

$$w_{klf} \geq z_f - (1 - i_{klf}) \qquad f \in F, k \in K_f, l \in L^+_{kf} \tag{2.20}$$

$$z_f \leq y_f \qquad f \in F \tag{2.21}$$

$$z_f \in [0,1] \qquad f \in F \tag{2.22}$$

$$n_k \in \mathbb{N}_0 \qquad k \in K \tag{2.23}$$

$$w_{klf} \in [0,1] \qquad f \in F, k \in K_f, l \in L^+_{kf} \tag{2.24}$$

$$(2.3)–(2.9).$$

The main differences from the DFRLP are the introductions of the variable n_k, which represents the number of charging poles installed at a location k, and the variable z_f, which is the proportion of a flow f that can be covered. This is necessary, because some flows might not be able to be fully covered, due to the limited capacity of each charging station (cf. Staněk et al. 2023).

The objective function (2.13) maximizes, analogous to the previous models, the covered flow volume; with the only difference that the binary variable y_f is now the real-valued variable z_f. Constraint (2.14) ensures that exactly S charging poles are allocated. At least one charging pole needs to be placed if a charging station is opened at a location, which is ensured by (2.15). Constraint (2.16) limits the number of charging poles at each location to the value of M_k.

Constraint (2.17) ensures that the capacity of each charging station is not exceeded. The constraints (2.18) – (2.20) are linking constraints for the linearization variable $w_{lkf} := i_{lkf} z_f$. Constraint (2.21) links z_f and y_f and assures that if a flow is not covered at all ($y_f = 0$), no proportional coverage of the flow is possible. Constraint (2.17) is a linearization of the following non-linear constraint:

$$\sum_{f \in F:\ k \in K_f} \left(\sum_{l \in L^-_{kf}} t_{lk} i_{lkf} + \sum_{l \in L^+_{kf}} t_{kl} i_{klf} \right) v_f e_f \cdot z_f \leq Cap \cdot n_k \quad k \in K \tag{2.25}$$

The left side of this constraint represents the total energy demand at each potential charging facility. For each flow f that uses a certain facility location k, the energy demand at location k is calculated by summing the distances of the cycle segments that are used to get to this charging facility, once on the forward and once on the backward journey, multiplied by v_f, e_f and z_f. Therefore, the energy demand is linearly proportional to the distance travelled. The right side represents the corresponding total capacity of the facility (cf. Staněk et al. 2023).

The refueling frequency e_f is crucial to correctly model short trips where the range exceeds the round trip distance ($R > 2 \cdot \tilde{t}_{O_f D_f}$). If $e_f < 1$ for a flow, which means that the range of a vehicle on this flow is greater than the round trip distance, then no full charge is necessary. This means that not all vehicles of this flow need to stop at the charging facility, so the energy demand is reduced by the parameter e_f. If $e_f = 1$, the vehicle range is equal to or smaller than the round trip distance, and more than one charging stop per round trip is necessary to avoid running out of fuel (cf. Staněk et al. 2023).

2.5 Capacitated Minimum Flow Volume Coverage DFRLP (C+MC-DFRLP)

In the C+MC-DFRLP, the number of charging poles to allocate no longer needs to be predefined, instead the number of charging poles is minimized while achieving a prespecified coverage level using the parameter C (cf. Staněk et al. 2023).

Additional decision variables:

- n_k, z_f As described in section 2.4

Additional parameters:

- $C \in [0, 1]$ As described in section 2.2
- Cap, e_f, M_k As described in section 2.4

Model:

$$\min \sum_{k \in K} n_k \tag{2.26}$$

$$\text{s.t.} \qquad \frac{\sum_{f \in F} v_f z_f}{\sum_{f \in F} v_f} \geq C \tag{2.27}$$

$$(2.15) - (2.24)$$

$$(2.3) - (2.9).$$

In contrast to the C-DFRLP, the objective function (2.26) minimizes the total number of charging poles, while constraint (2.27) assures that at least the proportion C of the total flow volume is covered. Constraints (2.15) – (2.24) are adopted from the C-DFRLP, and (2.3) – (2.9) are again adopted from the DFRLP.

2.6 Combination of C+MC-DFRLP and LC-DFRLP (C+LC-DFRLP)

The aim of this model is to combine the model with location-dependent costs (LC-DFRLP) and the capacitated model (C+MC-DFRLP), where the number of charging poles acts as a decision variable. Instead of minimizing the number of overall charging poles, this model seeks to minimize the total cost. In contrast to the LC-DFRLP, in this model, the costs of each charging facility are not only influenced by its location, but also the number of installed charging poles. This approach is realized by splitting the costs into *fixed* and *variable* components. The *fixed cost* represent factors such as land prices and the costs of establishing an electricity supply. The *variable cost* reflect the costs of each installed charging pole.

To achieve the maximum coverage with minimal cost, the model employs a two-step approach: Initially, the covered flow volume is maximized. Subsequently, the total cost is minimized by assuring that the previously maximized flow volume coverage remains covered.

2.6.1 Step 1: Maximize Covered Flow Volume

This model closely resembles the C-DFRLP, with the key difference that the total number of charging poles does not need to be specified and the available budget can be set. To determine the maximum possible flow volume coverage, the budget can be set to a sufficiently high value, effectively removing budget limits, or the budget constraint can be eliminated altogether.

Additional parameters:

- $c_{v,u}, c_{v,s}, c_{v,r} \in \mathbb{R}_0^+$ Variable costs of each station type – the costs per installed charging pole at each of the 3 location types *urban, suburban* and *rural*
- $c_{f,u}, c_{f,s}, c_{f,r} \in \mathbb{R}_0^+$ Fixed costs of each station type – the costs for building a charging station at each of the 3 location types *urban, suburban* and *rural* (excluding the costs of the charging poles)
- $u_k, s_k, r_k \in \{0, 1\}$ Binary parameters, indicating whether station k is of type urban (u_k), suburban (s_k) or rural (r_k)
- $\mathrm{B} > 0$ As described in Section 2.3

Model:

$$\max \sum_{f \in F} v_f z_f \tag{2.28}$$

$$\begin{aligned}
\text{s.t.} \quad & \sum_{k \in K} u_k x_k c_{f,u} + u_k n_k c_{v,u} \\
& + s_k x_k c_{f,s} + s_k n_k c_{v,s} \\
& + r_k x_k c_{f,r} + r_k n_k c_{v,r} \leq B \qquad (2.29) \\
& (2.15) - (2.24) \\
& (2.3) - (2.9).
\end{aligned}$$

The covered flow volume is maximized by the objective function (2.28). Constraint (2.29) ensures that the total cost do not exceed the value of the budget parameter B, by summing the fixed costs and the variable costs of each station, depending on the location type. The binary parameters u_k, s_k and r_k determine the location type of each charging facility k, with each facility assigned a unique type. For any facility k, only one of u_k, s_k and r_k is set to 1 and the remaining two parameters are set to 0, based on the classification of locations. Constraints (2.3) – (2.9) are the same as in the DFRLP and (2.15) – (2.24) remain the same as in the C-DFRLP.

The outcome of this model denotes the maximum possible flow volume that can be covered given the current parameters. To calculate the parameter C, the result is expressed as a fraction of the total flow volume:

$$C = \frac{\max \sum_{f \in F} v_f z_f}{\text{Total Flow Volume (TFV)}} \tag{2.30}$$

This parameter C is required for minimizing cost in the subsequent step of the model.

2.6.2 Step 2: Minimize Cost

In this model, the total cost of allocating charging facilities is minimized while ensuring that at least a proportion C of the total flow volume is covered. The parameter C, is now used as an input parameter to minimize the costs while covering the maximum possible flow volume.

Additional parameters:

- C As described in section 2.2

Model:

$$\min \sum_{k \in K} u_k x_k c_{f,u} + u_k n_k c_{v,u} + s_k x_k c_{f,s} + s_k n_k c_{v,s} + r_k x_k c_{f,r} + r_k n_k c_{v,r} \tag{2.31}$$

$$\text{s.t.} \qquad \frac{\sum_{f \in F} v_f z_f}{\sum_{f \in F} v_f} \geq C \tag{2.32}$$

$$(2.15)\text{–}(2.24)$$

$$(2.3)\text{–}(2.9).$$

Analogous to the previously used budget constraint, the objective function (2.31) minimizes the total cost by summing the fixed costs and the variable costs of each opened station, depending on the location type. Constraint (2.32) enforces, analogous to constraint (2.11) of the MC-DFRLP, that the proportion of covered flow volume is greater than or equal to the parameter C. As in Step 1 of the model, constraints (2.3) – (2.9) are identical to those in the DFRLP and (2.15) – (2.24) remain the same as in the C-DFRLP.

2.6.3 Scalability and Computational Efficiency Analysis

The solve times[3] for both steps of the model – maximizing coverage and minimizing cost – were assessed across the four benchmark datasets `s40w20`, `s60w30`, `s80w40`, and `s100w50`. The characteristics of these datasets are described in Section 4.2, the hardware and software setup is outlined in Section 4.1, and detailed numerical results can be found in Chapter 5. Table 1 and Figure 2 display the solve times for each dataset, indicating the computational demand for both model steps.[4]

Table 1: Solve Time of Both Steps of the Model on the Full Dataset

Dataset	**Max. Coverage Solve Time (secs.)**	**Min. Cost Solve Time (secs.)**
`s40w20`	27.64	60.92
`s60w30`	284.56	864.41
`s80w40`	6 304.28	20 477.58
`s100w50`	40 168.41	357 495.64

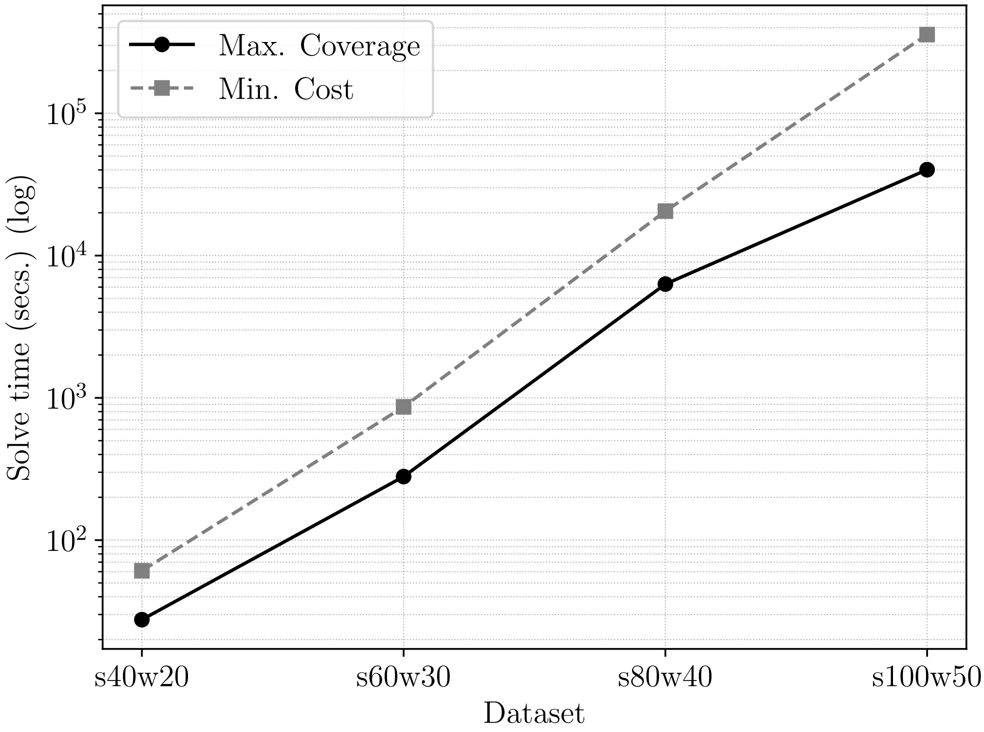

Figure 2: Logarithmic Plot of the Solve Time for the Datasets

The observed data demonstrates a strong increase in solve time with the growth in dataset

[3] time until the solver reaches a MIP Gap of 0%

[4] For the datasets `s40w20`, `s60w30` and `s80w40`, the median solve times from three solver runs are used. Due to the extensive solve time required for `s100w50`, only one run was feasible within reasonable time constraints.

size, particularly in the minimizing cost step. The solve time for `s100w50`, which contains 100 potential facility locations, is about 1450 times that of `s40w20` for maximizing the coverage, and about 5870 times that of `s40w20` for minimizing the cost, equating to approximately 4 days. This dramatic escalation of solve times, especially evident when visualized on a logarithmic scale (Figure 2), suggests an exponential growth in computational demands, since the runtime increase looks linear on the logarithmic scale. Based on this visual and quantitative analysis, the scalability of the models is considered questionable. This makes the model impractical for real-life applications, which are expected to involve much larger numbers of potential charging facility locations than the datasets tested.

While the initial solve time analysis highlighted significant efficiency challenges, several strategies were explored to address these issues. The following section details these attempts, setting the stage for the eventually successful strategy outlined in Chapter 3.

2.6.4 LP-based solution attempts

During the development of efficient computational strategies for the model, several less conventional approaches were attempted. These methods did not yield improvements in solve times or model performance, nor did they provide substantive insights into the problem structure. Nevertheless, they are documented here to illustrate the exploratory nature of this research and to acknowledge the discovery of a software bug during these trials. The approaches described in this section focus exclusively on modifying the Linear Program (LP) models.

2.6.4.1 Quadratic range constraint

One such approach involved transforming the range constraint into a quadratic form. This strategy was more experimental, exploring the solver's capacity for automatic linearization to verify if a different formulation could enhance runtime efficiency:

Updated range constraint:

$$\sum_{l \in L^{+}_{kf}} i_{klf} t_{klf} y_f \;\leq\; R \quad f \in F, k \in \{O_f \cup K_f\} \tag{2.33}$$

This range constraint is intended to be used instead of (2.3) and can therefore be used with both steps of the C+LC-DFRLP. Although the used solver (Gurobi 10.0.0) solved all of the benchmark datasets within seconds and reported optimal solutions, it yielded poor solution quality, especially for the cost-minimization step. In contrast, using the CPLEX solver yielded the correct optimal results previously found, albeit with significantly longer solve times. This led to collaboration with the Gurobi support, where a presolve-related bug was identified. This issue was fixed in version 10.0.1 (cf. Gurobi Optimization 2024). Using the updated version of Gurobi, the runtime returned to the less efficient performance levels observed before.

2.6.4.2 Alternative linearization of the range constraint

After the quadratic reformulation of the range constraint failed to reduce solve times, another linearization method was investigated. This approach was taken to assess whether the encountered inefficiencies are linked to the original linearization in constraint (2.3), which utilizes the Big M method. The key distinction lies in the usage of the binary auxiliary variable j_{klf}:

Additional decision variable:

- $j_{klf} \in \{0,1\}$ Auxiliary variable for the linearization of $i_{klf} \cdot y_f$

The following constraints replace constraint (2.3) in both steps of the C+LC-DFRLP (see Section 2.6.1 and Section 2.6.2):

Updated range constraints:

$$j_{klf} \leq i_{klf} \qquad f \in F, k \in L_f, l \in L_{kf}^{+} \tag{2.34}$$

$$j_{klf} \leq y_f \qquad f \in F, k \in L_f, l \in L_{kf}^{+} \tag{2.35}$$

$$i_{klf} + y_f - 1 \leq j_{klf} \qquad f \in F, k \in L_f, l \in L_{kf}^{+} \tag{2.36}$$

$$t_{kl} \cdot j_{klf} \leq R \qquad f \in F, k \in L_f, l \in L_{kf}^{+} \tag{2.37}$$

$$j_{klf} \in \{0,1\} \qquad f \in F, k \in L_f, l \in L_{kf}^{+} \tag{2.38}$$

The auxiliary variable j_{klf} is used to link the variables i_{klf} and y_f. Constraints (2.34) – (2.36) establish this relationship, by enforcing that j_{klf} is 1 if and only if both i_{klf} and y_f are 1. Constraint (2.37) is the linearized range constraint, ensuring that the driving range of the vehicles is not exceeded. Constraint (2.38) defines the auxiliary decision variable.

Utilizing this different linearization approach, no significant change in runtime was observed, suggesting that more complex approaches might be necessary to effectively decrease the solve time of the model. To maintain the model's conciseness, this linearization was discarded and the initial Big M method was used again from this point forward.

3 Decomposition method

Decompositions are strategies based on the *divide and conquer* approach. This involves splitting a complex problem into smaller subproblems, solving each part independently, and then combining the results to obtain the overall solution. These techniques have been widely used in optimization problems, including methods such as linear decompositions, dynamic programming and constraint programming (cf. Maniezzo et al. 2021, p. 159).
Due to the graph-based nature of the DFRLP and its extensions, a decomposition of the graph into smaller subgraphs may result in a substantial decrease in runtime, given that the C+LC-DFRLP's runtime for smaller instances is manageable. To exploit this potential, a straightforward problem-specific decomposition method was devised. The objective was to ascertain whether this simpler method could approach an optimal solution and effectively scale with increasing instance sizes. Should this approach prove insufficient, more sophisticated methods like Dantzig-Wolfe (see Dantzig and Wolfe 1960) and Bender's decomposition (see Benders 1962), which are renowned for efficiently decomposing linear problems into smaller subproblems, may be necessary.
This chapter outlines the development process and ideas that led to the final decomposition method, and details the devised decomposition method.

3.1 Development Process and Initial Approaches

The initial idea for a decomposition approach was to divide the graph into a grid-like structure (e.g., 2×2). The goal was to quickly reduce the complexity of the problem by creating smaller subproblems within each grid cell. However, this method has its limitations. The segmentation of the graph leads to a considerable number of flows being split between the grid cells. Several approaches could be possible: One solution would be to exclude flows that do not lie entirely within a single grid element. This would already affect a large number of the flows in the available datasets. An alternative would be to manipulate the flows such that only the parts of the flows that lie in the cluster are taken into account. This would necessitate an adjustment to the model formulation or a transformation of the data, as the formulation of the DFRLP only allows the consideration of entire flows and the origin and destination nodes are not in the same grid element in this case. Consequently, this approach was discarded.

To address the issue of non-separability of flows, the next approach of a decomposition method is based directly on the manipulation of the set of flows used. Instead of splitting the graph directly and thereby separating flows, the graph is separated by specifically removing flows from the set of flows. This does not require any reformulation of the model itself.

The smallest flow is removed from the graph until the graph breaks down into several clusters. This process is repeated until the size of the largest cluster falls below a predefined threshold. Then, both steps of the C+LC-DFRLP are applied to each of these resulting clusters consecutively. Initially, the objective is to maximize the coverage and subsequently, the objective is to minimize the total cost. The information about the stations that are ultimately opened in the cost-optimal solutions of the clusters can be used to determine the overall solution. In this first approach, the hypothesis was that charging stations that are not opened in the optimal solutions of the clusters are also unlikely to be opened in the optimal solution of the entire data set. This was implemented by fixing the values of x that were not opened in the clusters and then solving the model on the full dataset. Although this approach did not yield the desired results by far, it resulted in a significant reduction in runtime. This provided valuable insights for the subsequent development of the final decomposition, which is described in detail in the following section.

3.2 The proposed decomposition method

Figure 3 outlines the devised decomposition method:

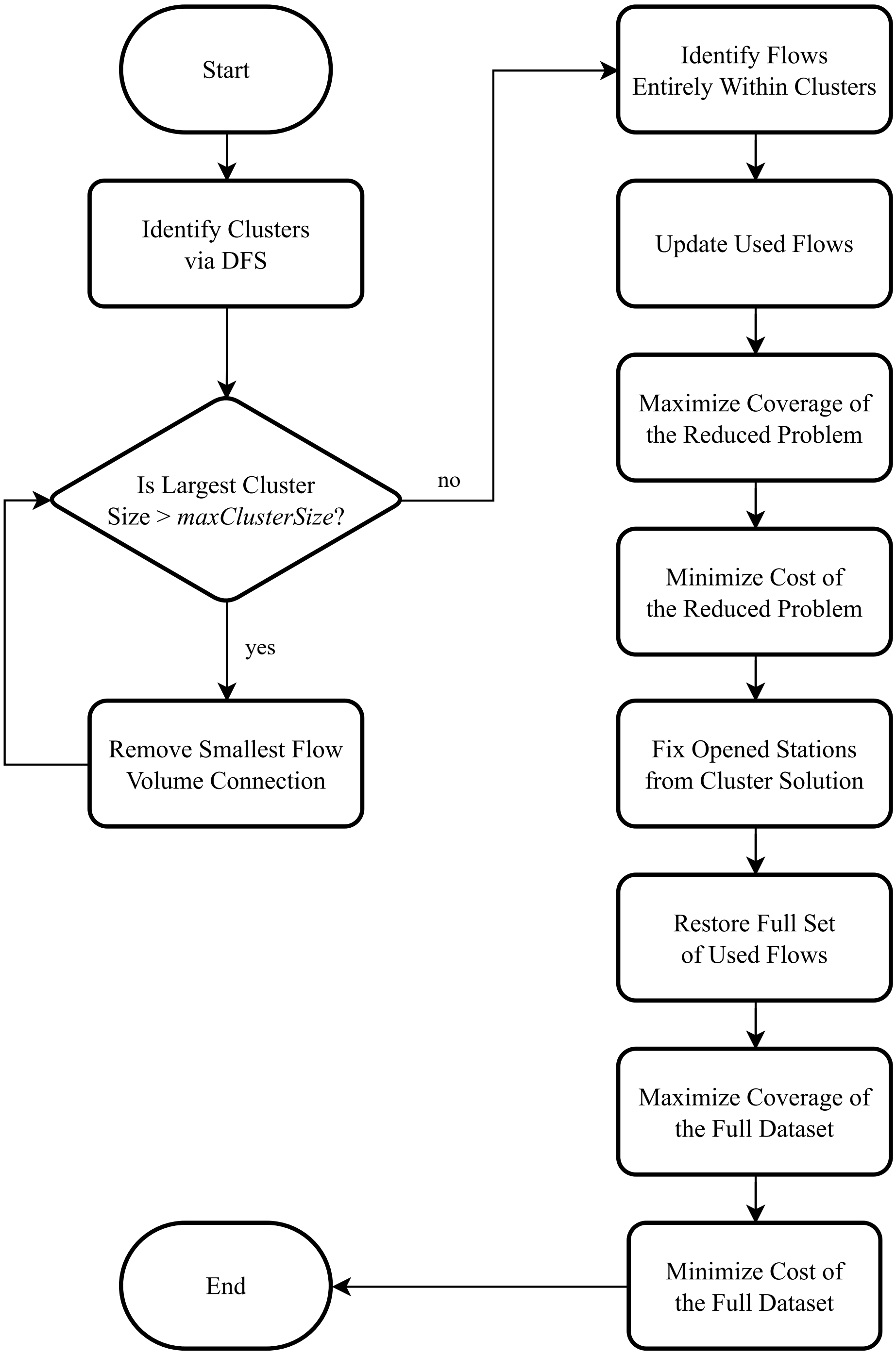
Start
Identify Clusters via DFS
Is Largest Cluster Size > *maxClusterSize*?
yes
no
Remove Smallest Flow Volume Connection
Identify Flows Entirely Within Clusters
Update Used Flows
Maximize Coverage of the Reduced Problem
Minimize Cost of the Reduced Problem
Fix Opened Stations from Cluster Solution
Restore Full Set of Used Flows
Maximize Coverage of the Full Dataset
Minimize Cost of the Full Dataset
End

Figure 3: Flowchart of the Decomposition Approach

- *Identify Clusters via DFS*: The algorithm starts by performing Depth-First Search (DFS) on the graph representing the dataset. This process identifies clusters that are already present in the dataset. Two nodes in the graph are considered connected, if there exists a connection between them with a flow volume greater than zero. A connection is considered to be an edge in the graph, and multiple flows can share one connection. Clusters consisting of only one node are eliminated, since they have no flows that connect them to the rest of the graph, making them irrelevant for the decomposition and the flow-based DFRLP in general.

- *Is Largest Cluster Size > maxSize?*: The core of the decomposition algorithm operates in a loop. The loop is only executed if there is at least one cluster that exceeds the maximum cluster size, referred to as *maxSize*. In this context, *size* is understood to refer to the cardinality, or the number of nodes, in the cluster. The cluster size is determined by DFS, which starts at a random location in the graph and traverses the graph until every connected node is visited.

- *Remove Smallest Flow Volume Connection*: Entering the loop, the connection with the smallest flow volume of the largest cluster is selectively removed from the graph. This approach is adopted for its simplicity and effectiveness, as it directly targets the connections that contribute the least to the overall problem. This process is repeated until no cluster has a size greater than *maxSize*.

- *Identify Flows Entirely Within Clusters*: Subsequently, all the flows that are completely within one of the different clusters (i.e. all locations of the flow are part of the cluster) are identified.

- *Update Used Flows*: The model formulation allows for the limitation of the model's scope by restricting the set of flows used, so in this step, the set of used flows is updated to only include the previously identified flows that are fully within the different clusters. This leads to a reduced problem instance which contains the different clusters. The structure of the resulting problem is similar to a typical structure found in *multiplant* models, known as *block angular* structure – these types of models are characterized by common rows (e.g. the objective function and one linking constraint, like the budget constraint in the C+LC-DFRLP) and independent blocks of coefficients, the submodels (cf. Williams 2013, pp. 45–49). Compared to the initial version of the decomposition method in Section 3.1, the different subproblems are solved with one run of the solver.

- *Maximize Coverage of the Reduced Problem*: Step 1 of the C+LC-DFRLP is applied to this reduced problem instance. The objective is to maximize the coverage of the flows within the clusters.

- *Minimize Cost of the Reduced Problem*: Using the coverage level Cdetermined in the previous step, the cost are minimized while maintaining the same level of coverage.

- *Fix Opened Stations from Cluster Solution*: Each charging station that is opened (where $x = 1$) in the cost-optimal solution of the clusters is fixed after this step. This is the opposite approach of what was previously attempted (see Section 3.1), and was the key difference in the development process that allowed the achievement of acceptable solutions.

- *Restore Full Set of Used Flows*: The original set of flows is restored, returning the problem to its full scope.

- *Maximize Coverage of the Full Dataset*: The coverage is maximized for the entire dataset, using the fixed stations from the cluster solutions.

- *Minimize Cost of the Full Dataset*: Finally, the costs are minimized, ensuring that the previously maximized coverage level is maintained. If a budget value is set that is sufficiently small to restrict the maximum coverable flow volume, the final step of minimizing costs can be omitted, as the budget restriction is already accounted for when maximizing coverage.

Initial tests suggest that the decomposition method offers significant runtime improvements compared to solving the full-scale problem. A detailed runtime analysis comparing this decomposition method to the full-scale solution approach will be presented in Chapter 5.

4 Test Setup

This chapter outlines the experimental setup used to evaluate the performance of the proposed decomposition method for the C+LC-DFRLP. The setup includes the software and hardware environments, detailed descriptions of the benchmark instances utilized, and the parameters and evaluation methodology applied throughout the experiments.

4.1 Software and Hardware

In their paper, Staněk et al. (2023) used AMPL (cf. Fourer et al. 2003) to model the different linear programs. Consistent with this approach, AMPL was also used in this thesis, with the majority of the AMPL code adopted from the authors. In addition to AMPL, the AMPL Java API was employed to develop a Java project, which enables the more complex requirements of the decomposition algorithm[5]. This Java project was crucial for managing AMPL operations, preprocessing, and dynamically managing data, all essential for the decomposition process described in Chapter 3.
All experiments were performed on a machine running Windows 11, equipped with an Intel i5-8250U CPU and 32 GB of RAM. The version of the Gurobi solver, which was used for the experiments, was 11.0.1.

4.2 Benchmark Instances

The four different used benchmark instances `s40w20`, `s60w30`, `s80w40` and `s100w50` were randomly generated by de Vries and Duijzer (2017, p. 111), based on a method proposed by Capar and Kuby (2012, p. 627). A randomly generated instance with `X` potential facility locations and `Y` OD nodes is referred to as `sXwY`. The coordinates are uniformly generated within a Euclidean plane of size $[0, 1000]^2$, assuming Euclidean distances between nodes. The edges of the graph are generated by creating the minimum spanning tree, and adding `X` additional edges by connecting the `X` closest node pairs that were not already connected. The OD nodes are created by duplicating `Y` of the `X` nodes, and each

[5] Although the Java project provided the foundation for the realization of this thesis, it will not be detailed extensively here as its contributions, while necessary, are technically supportive in nature. The tasks performed by the Java project include analyzing the graph, managing the AMPL operations, facilitating the graph decomposition, and storing results effectively.

OD pair represents a flow $f \in F$. The shortest path for every flow is determined using Dijkstra's algorithm, which also identifies potential facility locations along the path and computes a distance matrix. Flow volumes v_f are then generated by assigning each OD node a random value from a uniform distribution and dividing this value by the travel distance. An indicator function is applied to set the flow volume to zero for any flow with a travel distance under 100 units, emphasizing the focus on longer routes. The final step normalizes these volumes, ensuring the total flow volume equals 10^6 (cf. de Vries and Duijzer 2017, p. 111).

Table 2: Characteristics of the Benchmark Instances (Staněk et al. 2023)

Test instance	`s40w20`	`s60w30`	`s80w40`	`s100w50`
$\|K\|$	40	60	80	100
$\|OD\|$	20	30	40	50
$\|F\|$	190	435	780	1 225
$\min\{t_{O_f D_f}\}$	49.43	8.08	11.48	11.56
$\max\{t_{O_f D_f}\}$	2 071.06	2 752.80	2 556.34	2 192.80
$\widetilde{T}$	838.47	1 152.22	767.06	782.73
$\overline{T}$	907.70	1 152.77	822.52	834.48
TFV	10^6	10^6	10^6	10^6

Table 2 details the features of the benchmark instances used in this thesis: `s40w20`, `s60w30`, `s80w40`, and `s100w50`. The columns $|K|$, $|OD|$, and $|F|$ indicate the number of potential facility locations, origin-destination nodes, and flows, respectively. The table also shows the minimum and maximum travel distances per flow, illustrating the range of distances in the network. Additionally, it provides the median $\widetilde{T}$ and average $\overline{T}$ travel distances, offering insights into the typical journey lengths. All instances have a standardized Total Flow Volume (TFV) of 10^6, as described before.

4.2.1 Determination of the charging pole capacity

In order to determine the capacities per charging pole (Cap), Staněk et al. (2023) first calculate the median energy demand across all possible charging station locations, assuming that a charging station is opened at each location. Charging stations with zero energy demand are excluded. The authors then divide this median by the maximum number of charging poles per location M, assuming that $M = 4$. For the purposes of this thesis, the values of M and Cap are adopted. Table 3 presents the capacity per charging pole for the different datasets.

Table 3: Charging Pole Capacity per Dataset (Staněk et al. 2023)

Test instance	*Cap*
`s40w20`	2 801.01
`s60w30`	3 362.01
`s80w40`	1 678.57
`s100w50`	1 282.72

4.2.2 Classification of the location types

To classify the different nodes into the cost categories *urban*, *suburban* and *rural*, the density-based clustering algorithm used by Kastner et al. (2023, pp. 19–20) is adopted. Two criteria define the type of each node: the number of other nodes within a defined radius, and the amount of flow volume that starts or ends at this node. The clustering algorithm is outlined in Figure 4.

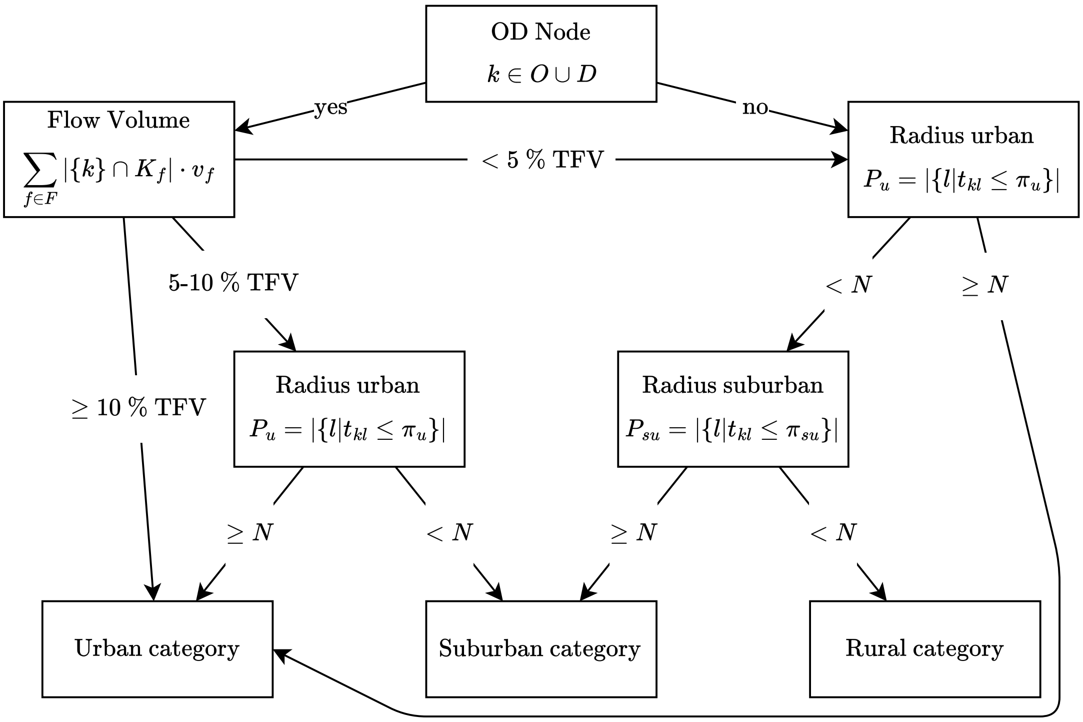

Figure 4: The Classification of Locations Into Different Categories, Redrawn From Kastner et al. (2023, p. 20)

Three parameters determine the algorithm's classification of location types: the density threshold N, the urban radius π_u and the suburban radius π_{su}. The chosen values for these parameters, as well as the number of locations that were assigned each location type, are outlined in table 4.

For nodes other than OD nodes, classification solely relies on the number of locations

Table 4: Location Type Parameters (Kastner et al. 2023, p. 21)

Test instance	`s40w20`	`s60w30`	`s80w40`	`s100w50`
N	4	4	4	5
π_u	100	100	100	100
π_{su}	150	150	150	150
urban (#)	13	9	24	18
sub-urban (#)	11	27	28	38
rural (#)	16	24	28	48

within the specified radii. If a node has at least N neighbors within the urban radius π_u, it is classified as *urban*. Otherwise, it is checked if it has at least N neighbors within the suburban radius π_{su}. If so, it is classified as *suburban*. Finally, if the node lacks N neighbors within either radius, it is classified as *rural*.

For OD nodes, an additional metric, the Total Flow Volume (TFV), is used for classification. If the sum of all flows involving this node meets or exceeds 10% of the TFV, the node is classified as *urban*. Flows between 5–10% of the TFV trigger another check: if at least N neighbors exist within the urban radius of the node, it remains *urban*. Otherwise, it becomes *suburban*. Finally, flow volumes below 5% of TFV rely solely on the number of neighbors in its urban and suburban radii for classification.

4.3 Parameters and Evaluation Layout

This section discusses the parameters influencing the model's performance, including solution quality and runtim. It also outlines the evaluation methodology and reviews the challenges, assumptions and limitations of this work. This sets the stage for the numerical evaluation presented in Chapter 5.

4.3.1 Cost parameters

Table 5 summarizes the values for the cost parameters:

Table 5: Costs by Location Type

Parameter Description	Location Type	Symbol	Value
Fixed Cost	Rural	$c_{f,r}$	1
	Suburban	$c_{f,s}$	2
	Urban	$c_{f,u}$	3
Variable Cost	Rural	$c_{v,r}$	1
	Suburban	$c_{v,s}$	1
	Urban	$c_{v,u}$	1

The assumptions behind the choice of the cost parameter values are discussed in more detail in Section 4.4. The fixed costs for rural, suburban, and urban locations are assigned values of 1, 2, and 3, respectively. This should reflect the higher land costs associated with higher urbanization. Similarly, a value of 1 is chosen for the variable cost, which accounts for the installation expense of a charging pole.

4.3.2 General Model Parameters

Table 6 lists the general model parameters:

Table 6: General Model Parameters

Parameter Description	Symbol	Value
Driving Range	R	250
Maximum number of charging poles	M_k	4
Charging Pole Capacity	Cap	see Table 3

The parameter values for the driving range R, the maximum number of charging poles M_k and the charging pole capacity Cap have been adopted from Staněk et al. (2023).

4.3.3 Clustering-Specific Parameter

Table 7 displays the parameter value relevant for the decomposition:

Table 7: Decomposition Parameters

Parameter Description	Value
Maximum Cluster Size	20

For the decomposition method, the Maximum Cluster Size is crucial, as it determines the scale of the problem instances that the solver needs to solve. To determine an appropriate value for this parameter, values ranging from 10 to 40, with a step size of 2, were evaluated. The total runtime[6] in seconds vs. the Maximum Cluster Size for the different benchmark instances are presented in Figure 5.

[6] The total runtime consists of the solve times for: Maximize Coverage (Cluster only) + Minimize Cost (Cluster only) + Maximize Coverage (Full Dataset) + Minimize Cost (Full Dataset). For the graphs, the median solve times of three runs are used. The calculation of the solve time is detailed in the beginning of Chapter 5

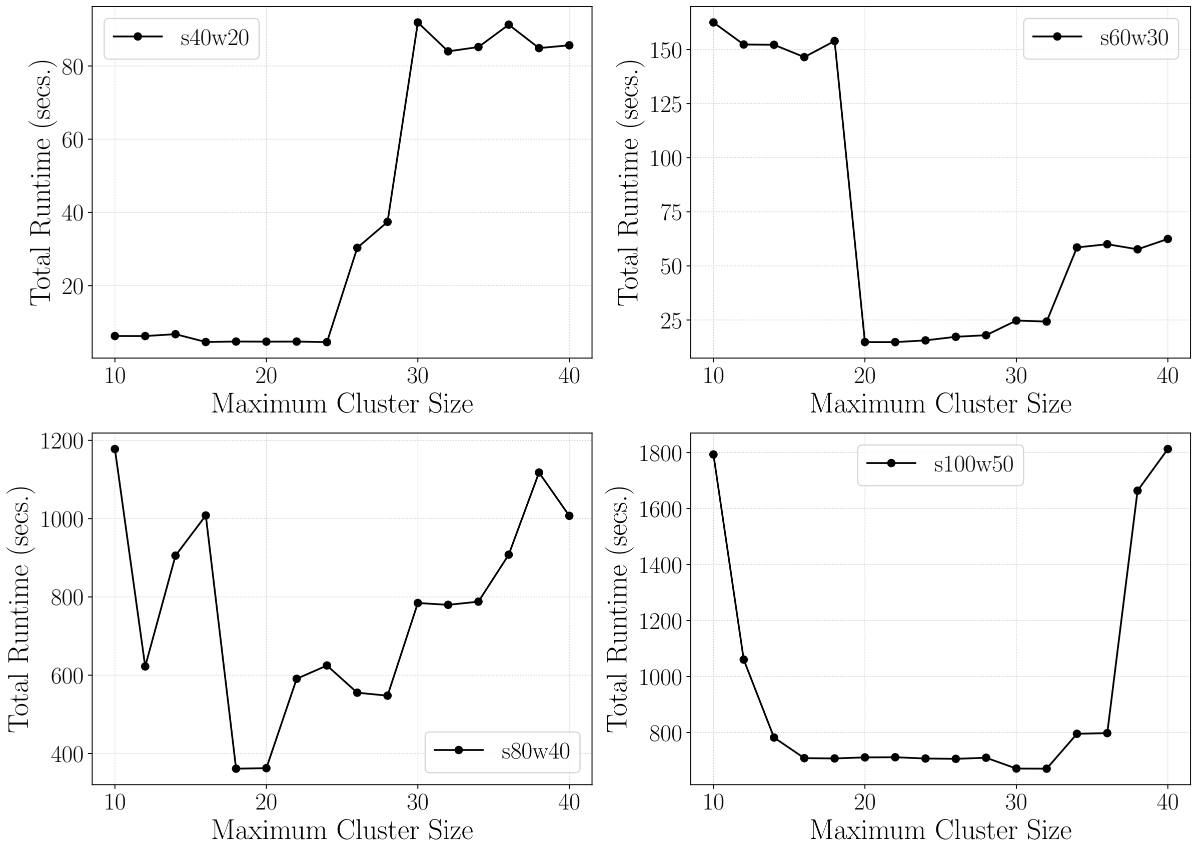

Figure 5: Total Runtime Dependent on the Maximum Cluster Size

The data in the different graphs reveal that the runtime depends on the Maximum Cluster Size. Except for the smallest dataset `s40w20`, it can be observed that the total runtime for either very small cluster sizes or cluster sizes over 30 increases substantially. While the graph for `s100w50` resembles a bathtub curve, the graph for the dataset `s80w40` does not show the same distinct pattern: instead, there are several plateaus and spikes, where the specific value of Maximum Cluster Size leads to an unfavorable solve process.

Across the different datasets, a value of 20–28 leads to advantageous total runtimes. Because not only the runtime, but also the solution quality is affected by the value Maximum Cluster Size, Figure 6 shows which values for Max. Coverage and Min. Cost can be achieved.

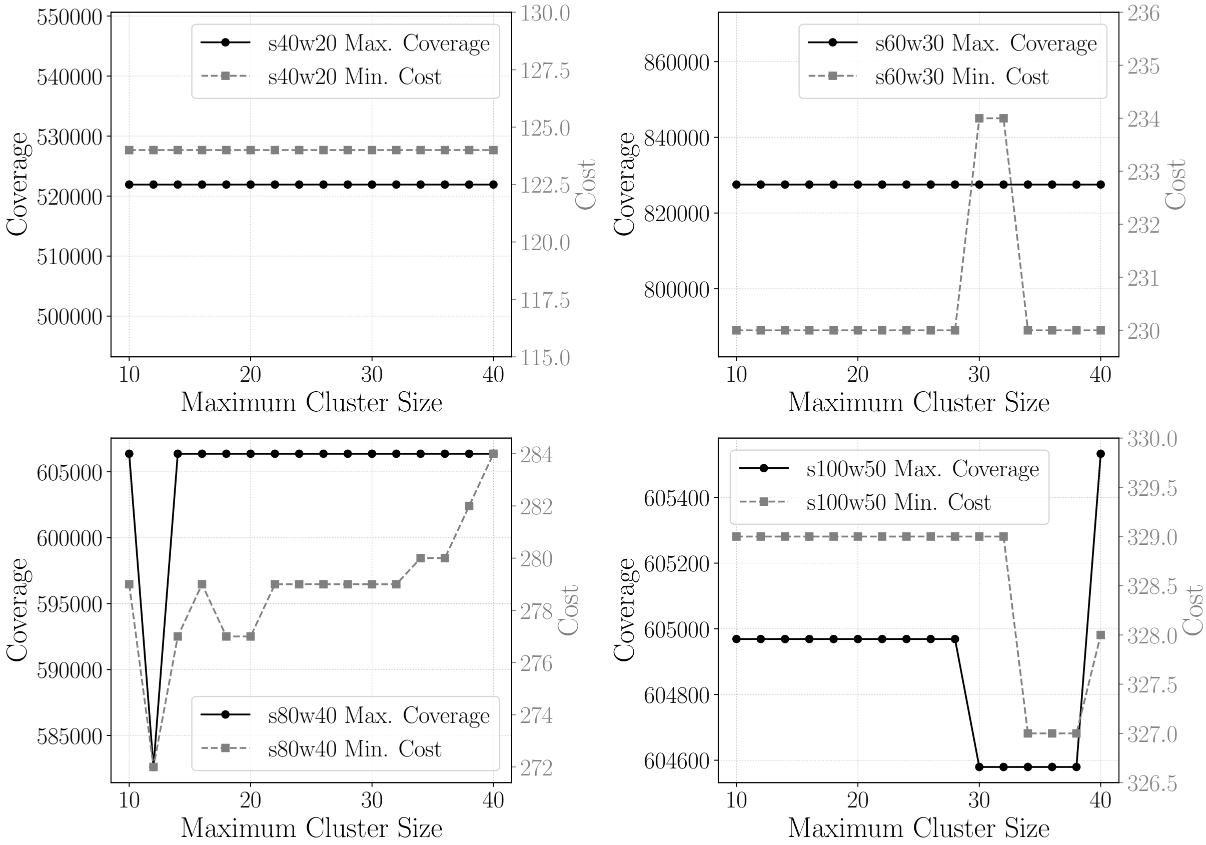

Figure 6: Max. Coverage and Min. Cost Dependent on the Maximum Cluster Size

The plots in Figure 6 depict the relation between the objective functions of both steps of the C+LC-DFRLP and the Maximum Cluster Size. Each graph has two y-axes: one for the coverage, depicted in black, and one for the cost, depicted in gray. The lines representing the corresponding values are colored accordingly.

While the coverage and the cost for `s40w20` and `s60w30` remain stable and nearly stable over the entire range of the tested values, the Max. Coverage and Min. Cost levels of `s80w40` and `s100w50` show some fluctuation. Based on these observations, a Maximum Cluster Size of 20 is chosen going forward.

4.4 Challenges, Assumptions and Limitations

- It is assumed that for all stations with the same location type, the fixed and the variable costs are the same. The model would require reformulation in order to allow for different costs for locations that share the same location type. In real-world scenarios, there can be substantial costs differences within each location type.
- Another assumption is, that both fixed and variable costs are one-off cost. With the current formulation, it is not possible to factor in time-dependent cost, e.g., the leasing of a property to build a charging station on or maintenance cost.

- Several additional assumptions have been made for values of the cost parameters. The fixed cost values are based on the assumption that it is generally more expensive to build a charging station in urban areas than in rural areas, primarily due to the higher land costs associated with more densely populated locations. It can be argued that the costs of establishing an electricity supply could outweigh the land costs in rural settings – however, the parameter choice depends highly on the geographical area where the model should be applied. The focus of this thesis is on the algorithm's practicability rather than its application to specific geographic areas.
  The chosen value for the variable cost parameters is based on the assumption that the costs of installing an additional charging pole are the same in every type of location. It is not taken into account that the installation of an additional charging pole might require a larger plot of land, or additional transformers, which could be counted towards the variable cost and lead to different charging pole cost.

- A limitation of the present work is that it is not possible to make any statements regarding the performance of the decomposition algorithm with larger datasets. The decision was made to adhere to the same datasets used by Staněk et al. (2023) in order to facilitate a certain degree of comparability between the results. There are a number of datasets in the literature (cf. Kchaou-Boujelben 2021, p. 25) which could have been used with the present model after some transformation steps. Due to the suboptimal runtime behaviour of the C+LC-DFRLP applied to large datasets, it is unlikely that an optimal solution could be identified within a meaningful time frame to assess the performance of the decomposition algorithm.

4.5 Evaluation Layout

This section outlines the used methodology to assess the decomposition method's performance compared to the full dataset approach. A detailed evaluation is performed focusing on two main aspects:

- Solution Quality: The optimality of the solutions, assessing how far the solutions using the decomposition method deviate from the optimal solutions of the full dataset approach.

- Runtime: The time taken to solve both steps of the C+LC-DFRLP and obtain the Maximum Coverage or the Minimum Cost, depending on the choice of the budget value.

The process for evaluating the C+LC-DFRLP involves three steps:

1. Maximize achievable CFV: The objective of this step is to identify the maximum achievable CFV with the given data and parameters. This step is performed without constraining the budget, setting a benchmark for solution quality.
2. Minimize costs while maintaining CFV: In this step, the goal is to establish the lowest possible costs while maintaining the maximum achievable CFV. These costs are used for the subsequent step.
3. Maximize CFV with incremental budgets: This is achieved by varying the budget incrementally, which allows for the analysis of how the CFV can be maximized under different financial constraints.

All of these steps are performed using the full dataset and using the decomposition method. The findings from these evaluation are compiled and detailed in Chapter 5.

5 Numerical Results

This chapter presents and analyzes the results of the evaluation of the C+LC-DFRLP using both the full dataset and the decomposition method. The results are structured around two primary performance metrics: runtime and solution quality. Each metric is discussed in the context of the three main evaluation steps outlined in Section 4.5: maximizing achievable CFV, minimizing the cost while maintaining the maximum CFV, and maximizing CFV with incremental budgets.

For the sake of clarity and consistency, all numerical values (with the exception of solve times) are rounded to three decimal places. In the following analysis, the term *solve time* refers to the time required by the called solver to reach a MIP gap of 0% or the respective time limit. The solve time is calculated by adding AMPL's parameters `_ampl_elapsed_time` and `_solve_elapsed_time`, analogous to the approach taken by Staněk et al. (2023). All solve times, with one exception, are the median solve time from three solver runs. Only the solve times for the full dataset solutions of `s100w50` represent a singular solve time, due to the extensive solve time required. The parameter values used for the decomposition are described in Section 4.3.

5.1 Optimal solutions for the full dataset

In order to provide context for the rest of this chapter, this section presents the optimal solutions obtained by applying the C+LC-DFRLP to the full datasets. For each dataset, the maximum achievable Covered Flow Volume (CFV) was obtained by solving Step 1 of the C+LC-DFRLP without the budget constraint. Subsequently, Step 2 of the model was solved to determine the minimum budget required to attain the previously determined maximum CFV.

Table 8 presents the optimal objective values and the corresponding solve times for each of the full dataset solutions:

Table 8: Optimal Objective Values and Solve Times for Full Dataset Solutions

Dataset	Max. CFV	Solve Time (secs.)	Min. Cost	Solve Time (secs.)
s40w20	521 918.727	27.64	124	60.92
s60w30	827 524.963	284.56	230	864.41
s80w40	606 369.353	6 304.28	277	20 477.58
s100w50	605 922.376	40 168.41	340	357 495.64

The data provide a baseline for the evaluation of the decomposition method. By comparing the objective values and solve times with those obtained using the decomposition method, the effectiveness of the decomposition approach in reducing computational complexity and maintaining solution quality can be assessed.

5.2 Solution Quality Comparison

This section analyzes solution quality of the decomposition method by comparing it to the full dataset approach.

5.2.1 Maximize achievable CFV

As previously discussed, each dataset has an upper limit on the achievable CFV, determined by the chosen parameters. This maximum achievable CFV is found by performing Step 1 of the model without any budget constraints, either by setting an extremely high budget value or by removing the budget constraint entirely. Table 9 shows the objective values for Step 1 of the C+LC-DFRLP, comparing the results obtained from the full dataset with those from the decomposition method. This table highlights the relative difference in CFV between the two approaches:

Table 9: Objective Values for Step 1 of the C+LC-DFRLP

Dataset	Full Dataset Max. CFV	Decomposition Max. CFV	Relative Difference (%)
`s40w20`	521 918.727	521 918.727	0%
`s60w30`	827 524.963	827 524.963	0%
`s80w40`	606 369.353	606 348.668	0.003%
`s100w50`	605 922.376	604 968.941	0.157%

The data indicate that for the datasets `s40w20` and `s60w30`, the optimal solution is found using the decomposition method. For `s80w40`, the absolute difference in objective values is only 20.685, resulting in 0.003% relative difference. The objective value for `s100w50` shows the largest deviation from the optimum, with 0.157%.

5.2.2 Minimize Cost while maintaining maximum CFV

The previously determined CFV is now used for the parameter C in Step 2 of the model. This aims at determining the necessary budget to build enough charging stations to cover the maximum achievable CFV. Table 10 presents the minimum cost and the relative difference in cost between the two approaches:

Table 10: Objective Values for Step 2 of the C+LC-DFRLP

Dataset	Full Dataset Min. Cost	Decomposition Min. Cost	Relative Difference (%)
`s40w20`	124	124	0%
`s60w30`	230	230	0%
`s80w40`	277	277	0%
`s100w50`	340	329	3.235%

For the datasets `s40w20`, `s60w30` and `s80w40`, the same cost values are determined using the decomposition method as when applying the model on the full dataset. For the `s100w50` dataset, an objective value was identified represents a 3.235% improvement over the optimal solution. This outcome is a consequence of the lower CFV value that was identified in the preceding step (see Section 5.2.1).

5.2.3 Maximize CFV with Incremental Budgets

To enable a comparison of the efficacy of the decomposition method in the context of limited budget, several increments of the budget parameter are used – 100%, 75%, 50% and 25% of the minimum required budget for each dataset (see Table 8). Using these restricted budget values, Step 1 of the C+LC-DFRLP was performed to maximize the coverage. A time limit of 500 000 s was set. Table 11 summarizes the CFV outcomes for both the full dataset and the decomposition method:

Table 11: Objective Values for Step 1 of the C+LC-DFRLP With Incremental Budget Values

Dataset	Budget	Full Dataset CFV	Decomposition CFV	Relative Difference (%)
`s40w20`	124 (100% of 124)	521 918.727	521 918.727	0%
	93 (75% of 124)	494 427.352	488 755.738	1.147%
	62 (50% of 124)	399 951.168	399 951.168	0%
	31 (25% of 124)	203 070.249	203 070.249	0%
`s60w30`	230 (100% of 230)	827 524.963	827 524.963	0%
	172.5 (75% of 230)	765 819.825	760 651.535	0.675%
	115 (50% of 230)	622 114.885	585 132.908	5.945%
	57.5 (25% of 230)	352 217.693	323 949.994	8.026%
`s80w40`	277 (100% of 277)	606 369.353	606 348.668	0.003%
	207.75 (75% of 277)	566 153.440	561 835.851	0.763%
	138.5 (50% of 277)	448 065.997	434 172.794	3.101%
	69.25 (25% of 277)	277 293.489	262 922.167	5.183%
`s100w50`	340 (100% of 340)	605 922.376	604 968.941	0.157%
	255 (75% of 340)	576 373.271 [7]	572 998.559	0.586%
	170 (50% of 340)	459 310.241 [8]	444 577.400	3.208%
	85 (25% of 340)	278 824.228	251 709.818	9.725%

Across all datasets, the difference in the objective function is the lowest when 100% of the minimum required budget is available, with minimal relative differences (0% to 0.157%). This shows that the decomposition method provides the best results when the budget is not restricted. For the `s40w20` dataset, likely due to its size, the differences in objective values are the smallest, reaching optimal levels for every budget increment except 75%. The remaining datasets show an increasing relative objective value difference as the budget is reduced, with notable degradations especially at 25% budget, where the values range from 5.183% to 9.725%. The utilization of the decomposition method resulted in an average decrease in solution quality of 2.407%. The median difference across all budget increments is 0.719%.

[7] Relative MIP Gap: 1.355%, best bound: 584,184.329

[8] Relative MIP Gap: 6.465%, best bound: 489,005.130

For the different budget levels, the median and average differences in the objective value are summarized in Table 12:

Table 12: Summary of Differences in Objective Function by Budget Level

Budget Level	Median Relative Difference (%)	Average Relative Difference (%)
100%	0.002%	0.040%
75%	0.719%	0.793%
50%	3.155%	3.064%
25%	6.605%	5.734%
overall	0.719%	2.407%

This table presents a summary of the relative difference in objective values as a function of budget increments. The median as well as the average relative differences increase with a decreasing budget level, yet remain close to the optimal values. The overall median and average relative differences, across all budget increments, remain relatively low, with 0.719% and 2.407%, respectively.

5.3 Runtime Comparison

After putting the objective values of applying the C+LC-DFRLP in relation to the objective values when using the decomposition method, this section analyzes the corresponding runtimes.

5.3.1 Maximize achievable CFV

Due to the fact that the decomposition approach consists of multiple solver runs, the decomposition solve time for this step of the evaluation consists of the sum of the solve times of the multiple solver runs that are necessary to obtain the maximum achievable CFV for the full dataset: maximizing the CFV for the clusters, minimizing the cost for the clusters, and finally maximizing the CFV for the full dataset with the fixed stations from the previous steps.

Table 13 depicts the solve time that is needed to find the maximum achievable CFV for the full dataset and for the decomposition approach:

Table 13: Runtime for Step 1 of the C+LC-DFRLP: Full Dataset vs. Decomposition Method

Dataset	Full Dataset Solve Time (secs.)	Decomposition Solve Time (secs.)	Relative Solve Time Decrease (%)
`s40w20`	27.64	4.67	83.10%
`s60w30`	284.56	12.66	95.55%
`s80w40`	6 304.28	303.94	95.18%
`s100w50`	40 168.41	268.81	99.33%

Across all datasets, a substantial decrease in solve time can be observed. The smallest relative difference in runtime is for the smallest dataset `s40w20`, with 83.10%. For the other datasets, the relative difference is larger, with its largest value for the largest dataset `s100w50`.

5.3.2 Minimize Cost while maintaining maximum CFV

As described in the previous section, the decomposition approach consists of multiple solver runs. Before this cost minimization can be performed, the previous steps of the decomposition process need to be traversed as well, in order to fix the stations that are built in the cluster solution and enable the efficient solving of Step 2 of the C+LC-DFRLP. The decomposition solve time in Table 14 only represents the singular solve time of Step 2 of the model, which is run after the previous steps. The total runtime to obtain these minimized cost values is therefore the sum of the decomposition solve times presented in Table 13 and the corresponding decomposition solve times in Table 14:

Table 14: Runtime for Step 2 of the C+LC-DFRLP: Full Dataset vs. Decomposition Method

Dataset	Full Dataset Solve Time (secs.)	Decomposition Solve Time (secs.)	Relative Solve Time Decrease (%)
`s40w20`	60.92	1.31	97.845
`s60w30`	864.41	6.78	99.215
`s80w40`	20 477.58	660.50	96.775
`s100w50`	357 495.64	488.80	99.863

The data show a substantial decrease in solve time across all datasets, ranging from 96.775% for `s80w40` to 99.863% for `s100w50`. The average solve time decrease is 98.424%.

5.3.3 Maximize CFV with Incremental Budgets

Finally, the runtime efficiency of the decomposition method is evaluated under varying budget constraints. Table 15 compares the solve times for maximizing CFV at different budget levels, highlighting the relative solve time for each dataset:

Table 15: Solve times for Step 1 of the C+LC-DFRLP With Incremental Budget Values

Dataset	Budget	Full Dataset Solve Time	Decomposition Solve Time	Relative Solve Time Decrease (%)
`s40w20`	124 (100% of 124)	27.64	4.67	83.10%
	93 (75% of 124)	98.33	5.96	93.94%
	62 (50% of 124)	91.41	8.98	90.17%
	31 (25% of 124)	93.11	15.63	83.22%
`s60w30`	230 (100% of 230)	284.56	12.66	95.55%
	172.5 (75% of 230)	4 614.27	17.89	99.61%
	115 (50% of 230)	2 428.14	124.20	94.88%
	57.5 (25% of 230)	976.73	57.53	94.11%
`s80w40`	277 (100% of 277)	6 304.28	303.94	95.18%
	207.75 (75% of 277)	251 864.08	2 028.81	99.19%
	138.5 (50% of 277)	208 807.86	838.11	99.60%
	69.25 (25% of 277)	115 459.55	1 931.02	98.33%
`s100w50`	340 (100% of 340)	40 168.41	268.81	99.33%
	255 (75% of 340)	500 056.77[9]	1 890.95	99.62%
	170 (50% of 340)	500 038.92[10]	970.22	99.81%
	85 (25% of 340)	422 145.89	725.20	99.83%

Across all datasets, the decomposition method consistently shows a significant reduction in solve time compared to running the model on the full datasets. For the smaller datasets `s40w20` and `s60w30`, the largest difference in solve time can be observed when the budget is constrained to 75% of the minimum required budget. For `s80w40`, the greatest improvement in solve time is observed at 50% of the minimum required budget, while for `s100w50`, the greatest improvement is observed at the 25% level. For `s100w50`, the solve processes at the 75% and 50% levels were interrupted after 500 000 s. Due to the significant MIP Gap of 6.465% at the 50% level in contrast to the MIP Gap of 1.355% at 75%, the total solve time for the 50% level is likely the greatest. This observation leads to the conclusion that the greatest difference in solve time for the `s100w50` dataset is likely at the 50% stage.

[9] Relative MIP Gap: 1.355%, best bound: 584,184.329

[10] Relative MIP Gap: 6.465%, best bound: 489,005.130

Table 16 summarizes the median and average solve time decreases for the different budget increments:

Table 16: Summary of Solve Time Differences by Budget Level

Budget Level	Median Solve Time Decrease (%)	Average Solve Time Decrease (%)
100%	95.366%	93.290%
75%	99.403%	98.093%
50%	97.242%	96.115%
25%	96.219%	93.871%
overall	96.940%	95.342%

In contrast to Table 12, the median and average solve time differences do not provide a clear picture of a consistent decline in performance with lower budget levels. Both the median and average peak at 75% of the budget, but it is also important to note that the values for 75% and 50% are influenced by the solve time limitation of `s100w50`. Without the previously mentioned solve time limitation, it can be assumed that both of these values would be higher than those depicted in the table above. The overall and median solve time over all budget increments and datasets are 96.940% and 95.342%, respectively, indicating a substantial runtime decrease.

6 Conclusions and Future Research

This thesis presents a novel extension of the DFRLP, the C+LC-DFRLP. The combination of the existing model extensions LC-DFRLP and C-DFRLP offers a more realistic modeling of real-world scenarios. However, the combination of these models led to unfavorable runtime behavior, which makes it unusable for real-world applications. To address this challenge, a straightforward problem-specific decomposition approach was developed and implemented, significantly reducing the runtime of the C+LC-DFRLP model and making it feasible for use with larger datasets.

To assess the performance of the decomposition approach in contrast to running the model on the full dataset, the solve time and the objective values of both approaches were compared on different artificial datasets. The evaluation focused on testing both steps of the model (maximize coverage and minimize cost) under varied conditions. The results demonstrated that the decomposition approach significantly reduces the solve time. For maximizing the coverage with different budget increments, the average solve time decreased by 95.342%, while the mean relative difference of the objective values was only 2.407%. In terms of minimizing the cost, the average runtime improvement is 98.424%, with the optimal solutions identified for three of the four benchmark datasets and a superior solution[11] for the largest one.

These findings underscore the potential of this straightforward decomposition method to achieve a balance between scalability and efficiency without significantly compromising on the quality of the solutions. This balance is particularly important for further extensions of the model that complicate the problem.

The enhancements to the DFRLP presented in this thesis contribute to the advancement of operations research applied to logistics and urban planning. This research paves the way for further investigation that could lead to more sophisticated, realistic, and practical solutions to complex logistical challenges.

Further research should test the performance of the decomposition method with larger, real-world datasets, to validate and potentially enhance its scalability. Additionally, exploring alternative subgraph generation methods, such as the minimum cut approach, could potentially improve performance.

As an extension of the DFRLP, the C+LC-DFRLP also inherits certain limitations. No-

[11] Due to a slightly inferior coverage value, which can be satisfied by constructing fewer charging stations.

tably, the assumption of static flow volumes over time restricts the model's applicability to dynamic real-world scenarios. As previously mentioned by Staněk et al. (2023), another potential area for improvement of the model lies in modelling stochasticity the driving range. The use of a fixed value for the driving range of all BEVs is a simplification of reality, where the driving range is stochastic. A wide range of different BEVs are available, each with varying driving ranges. Additionally, several factors such as driving style, air temperature and battery degradation influence the total driving range of the vehicles.

In summary, the research presented in this thesis demonstrates the feasibility and effectiveness of a decomposition method for the C+LC-DFRLP, significantly improving runtime efficiency while maintaining high solution quality. Future research should aim to build on these findings, enhancing the model's applicability and robustness in real-world scenarios.

Literature

Book Sources

Fourer, Robert, David M. Gay, and Brian W. Kernighan (2003). *AMPL: A modeling language for mathematical programming.* 2. ed. Pacific Grove, Calif.: Thomson Brooks Cole. ISBN: 978-0534388096

Lee, Hoesung et al. (2023). *IPCC, 2023: Climate Change 2023: Synthesis Report. Contribution of Working Groups I, II and III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Core Writing Team, H. Lee and J. Romero (eds.)]. IPCC, Geneva, Switzerland.* Intergovernmental Panel on Climate Change (IPCC). DOI: 10.59327/IPCC/AR6-9789291691647

Maniezzo, Vittorio, Marco Antonio Boschetti, and Thomas Stützle (2021). *Matheuristics: Algorithms and Implementations.* 1st ed. 2021. Springer eBook Collection. Cham: Springer International Publishing and Imprint Springer. ISBN: 978-3-030-70276-2. DOI: 10.1007/978-3-030-70277-9

Williams, H. Paul (2013). *Model Building in Mathematical Programming.* 5th ed. New York, NY: Wiley, J. ISBN: 9781118443330

Journal Articles

Benders, J. F. (1962). "Partitioning procedures for solving mixed-variables programming problems". In: *Numerische Mathematik* 4.1, pp. 238–252. ISSN: 0029-599X. DOI: 10.1007/BF01386316

Capar, Ismail and Michael Kuby (2012). "An efficient formulation of the flow refueling location model for alternative-fuel stations". In: *IIE Transactions* 44.8, pp. 622–636. ISSN: 0740-817X. DOI: 10.1080/0740817X.2011.635175

Dantzig, George B. and Philip Wolfe (1960). "Decomposition Principle for Linear Programs". In: *Operations Research* 8.1, pp. 101–111. ISSN: 0030-364X. DOI: 10.1287/opre.8.1.101

de Vries, Harwin and Evelot Duijzer (2017). "Incorporating driving range variability in network design for refueling facilities". In: *Omega* 69, pp. 102–114. ISSN: 0305-0483. DOI: 10.1016/j.omega.2016.08.005

Hodgson, M. John (1990). "A Flow–Capturing Location–Allocation Model". In: *Geographical Analysis* 22.3, pp. 270–279. ISSN: 0016-7363. DOI: 10.1111/j.1538-4632.1990.tb00210.x

Hosseini, Meysam and S. A. MirHassani (2017). "A heuristic algorithm for optimal location of flow–refueling capacitated stations". In: *International Transactions in Operational Research* 24.6, pp. 1377–1403. ISSN: 0969-6016. DOI: 10.1111/itor.12209

Kchaou-Boujelben, Mouna (2021). "Charging station location problem: A comprehensive review on models and solution approaches". In: *Transportation Research Part C: Emerging Technologies* 132, 103376, pp. 1–33. ISSN: 0968090X. DOI: 10.1016/j.trc.2021.103376

Kuby, Michael and Seow Lim (2005). "The flow-refueling location problem for alternative-fuel vehicles". In: *Socio-Economic Planning Sciences* 39.2, pp. 125–145. ISSN: 00380121. DOI: 10.1016/j.seps.2004.03.001

Lim, Seow and Michael Kuby (2010). "Heuristic algorithms for siting alternative-fuel stations using the Flow-Refueling Location Model". In: *European Journal of Operational Research* 204.1, pp. 51–61. ISSN: 03772217. DOI: 10.1016/j.ejor.2009.09.032

Staněk, Rostislav, Peter Greistorfer, and Anna Elisabeth Kastner (2023). "Advanced optimization models for the location of charging stations in e-mobility". In: *Central European Journal of Operations Research.* ISSN: 1435-246X. DOI: 10.1007/s10100-023-00878-w

Upchurch, Christopher, Michael Kuby, and Seow Lim (2009). "A Model for Location of Capacitated Alternative–Fuel Stations". In: *Geographical Analysis* 41.1, pp. 85–106. ISSN: 0016-7363. DOI: 10.1111/j.1538-4632.2009.00744.x

Wang, Ying-Wei and Chuah-Chih Lin (2013). “Locating multiple types of recharging stations for battery-powered electric vehicle transport”. In: *Transportation Research Part E: Logistics and Transportation Review* 58, pp. 76–87. ISSN: 13665545. DOI: 10.1016/j.tre.2013.07.003

Online Sources

Gurobi Optimization, LLC. (2024). *Recent Bug Fixes by Version: v10.0.1.*
https://www.gurobi.com/downloads/recent-bug-fixes-by-version/#GO1001
(visited on Apr. 16, 2024)

Kastner, Anna Elisabeth, Peter Greistorfer, and Rostislav Staněk (2023). *Advanced optimization models for the location of charging stations in e-mobility.*
https://arxiv.org/pdf/2109.14358.pdf
(visited on Feb. 22, 2024)

Appendix

To fulfill the requirements by the University of Leoben to declare the use of AI tools, Table 17 outlines the usage areas and the contribution of AI to this thesis:

Table 17: Declaration of the Usage of AI-based Tools

Subject	AI Contribution (%)	Tool/Version	Reference to prompting
Improvement of linguistic readability	30	ChatGPT-4 & Gemini 1.5 Pro & DeepL Write	no copyright-relevant prompts
Support in translation	15	DeepL	no copyright-relevant prompts
Documentation of the Java code	75	ChatGPT-4	no copyright-relevant prompts
Java code improvements	25	ChatGPT-4 & Gemini 1.5 Pro	no copyright-relevant prompts