TU WIEN Informatics

Knowledge Graph-Driven Tour Optimization for Sustainable Waste Collection

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Logic and Computation

eingereicht von

Adrian Bracher
Matrikelnummer 01637180

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Prof. Dr. Emanuel Sallinger
Mitwirkung: Dr. Markus Nissl

Wien, 20. März 2025

Adrian Bracher	Emanuel Sallinger

TU WIEN Informatics

Knowledge Graph-Driven Tour Optimization for Sustainable Waste Collection

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Logic and Computation

by

Adrian Bracher
Registration Number 01637180

to the Faculty of Informatics

at the TU Wien

Advisor: Prof. Dr. Emanuel Sallinger
Assistance: Dr. Markus Nissl

Vienna, March 20, 2025

Adrian Bracher	Emanuel Sallinger

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Adrian Bracher

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.
Ich erkläre weiters, dass ich mich generativer KI-Tools lediglich als Hilfsmittel bedient habe und in der vorliegenden Arbeit mein gestalterischer Einfluss überwiegt. Im Anhang „Übersicht verwendeter Hilfsmittel“ habe ich alle generativen KI-Tools gelistet, die verwendet wurden, und angegeben, wo und wie sie verwendet wurden. Für Textpassagen, die ohne substantielle Änderungen übernommen wurden, haben ich jeweils die von mir formulierten Eingaben (Prompts) und die verwendete IT- Anwendung mit ihrem Produktnamen und Versionsnummer/Datum angegeben.

Wien, 20. März 2025

Adrian Bracher

Acknowledgements

I would like to thank Prof. Dr. Emanuel Sallinger and Dr. Markus Nissl for their constructive feedback and guidance, which kept me motivated throughout this work. I'm also grateful to my colleague Jonathan Lex for his valuable input and collaborative efforts.

This work has been supported by the Vienna Science and Technology Fund (WWTF) under the project NXT22-018. [1]

[1] https://www.wwtf.at/funding/programmes/vrg/VRG18-013/

Kurzfassung

Aktuelle gesellschaftliche Herausforderungen wie der Klimawandel, die Verschmutzung der Ökosysteme unseres Planeten und die Abhängigkeit von knappen Ressourcen in einer politisch aufgeladenen Welt motivieren den Wandel hin zu einer Kreislaufwirtschaft, wobei Recycling eine essentielle Komponente für den Übergang darstellt. Recycling von organischen Feststoffabfällen wird durch die Verunreinigung mit schädlichen Materialen erschwert, die oft dazu führen, dass die Abfälle verbrannt oder auf Deponien entsorgt werden, wodurch die Treibhausgasemissionen steigen und wertvolle Ressourcen verloren gehen. Bei der Abholung solcher Abfälle kann stark verunreinigter Müll von einer einzigen Abholung eine ganze Müllwagen-Ladung kontaminieren. Gleichzeitig würde die Entsendung mehrerer Müllwagen für eine bessere Trennung der Abfälle die Betriebskosten und die durch den Transport verursachten Emissionen erhöhen. Daher stellen wir uns die Frage, ob eine Vorsortierung der Abholungen auf der Grundlage historischer Schadstoffmessungen den Gesamtaufwand verringern und die Recycling-Quote erhöhen kann?

In dieser Arbeit präsentieren wir einen Lösungsvorschlag, der Prognosemodelle und Tourenoptimierung in einem auf Knowledge Graphen basierenden Framework integriert. Dafür definieren wir dieses Problem als Pre-Collection Sorting Problem und erstellen eine problemspezifische Optimierungsontologie für die Abfallwirtschaft. Wir identifizieren zwei Strategien für die Prognose des Kontaminationslevels zukünftiger Abholungen: Klassifizierung von Haltestellen, und Modellierung von Abholungen derselben Haltestelle als Stichproben eines stochastischen Prozesses, der anhand früherer Verschmutzungsdaten parametrisiert wird. Für die Optimierung der Touren haben wir eine Greedy-Heuristik und einen lokalen Suchalgorithmus entwickelt, die auf das Problem und beide Strategien der Kontaminationsprognose zugeschnitten sind.

Die entwickelten Methoden werden in 11 Emissionsszenarien an einer Probleminstanz aus Echtdaten, die über sechs Monate erhoben wurden, getestet. Wir evaluieren die Einzelkomponenten separat und integriert als Gesamtlösung. Dabei bewerten wir die Vorhersagegenauigkeit, die Routeneffizienz und resultierende Umweltauswirkungen. Im Vergleich zur Ausgangslage der aus den Daten extrahierten Status-quo-Routen reduzieren wir in allen Szenarien bis zu 2/3 der entsorgten Abfallmenge. Darüber hinaus beobachten wir eine ähnliche Reduktion der Gesamtemissionen, was demonstriert, dass es sich um eine praxisnahe Lösung für diese wichtige Herausforderung der Abfallentsorgung handelt.

Abstract

Current societal challenges such as climate change, pollution of the planet's ecosystems, and the dependence on rare resources in a politically charged world motivate the change towards a circular economy, with recycling being one of the essential components to a successful transition. Organic solid waste recycling faces challenges due to contamination from harmful materials, often resulting in waste being incinerated or sent to landfills, which raises greenhouse gas emissions and wastes valuable resources. When collecting such waste, a single severely contaminated collection stop can spoil an entire truckload. Yet dispatching multiple trucks for separation increases operational costs and emissions from transportation. This raises the question whether an initial sorting phase, which involves sorting the stops according to pollution data, can reduce the total effort and improve the recycling quota?

In this thesis, we address this question by proposing a solution that integrates predictive modeling and tour optimization algorithms in a knowledge graph-based framework. For this purpose, we formalize the problem at hand as a Pre-Collection Sorting Problem and establish a problem-specific layered optimization ontology for the waste domain. We identify two strategies for predicting contamination levels of future collection events: Classification of tour stops, and modeling repeated visits to the same stop as samples from a stochastic process parameterized by previous pollution records. For the tour optimization task, we designed a greedy heuristic and a sophisticated local search algorithm tailored to the problem and both contamination prediction strategies.

The proposed methods are tested on 11 emission scenarios in a challenging real-world problem instance with waste collection data over six months. We evaluate the solution components separately and end-to-end, measuring prediction accuracy, route efficiency, and environmental impact. Compared to the baseline of status quo routes extracted from the data we reduce up to two thirds of disposed waste volume in all scenarios. Furthermore, we also observe a similar reduction of total greenhouse gas emissions, demonstrating a practical and sustainable solution for this real-world waste management challenge.

Contents

CHAPTER 1

Introduction

Sustainability in our economy is essential for preserving the balance of our planet's ecosystems. Traditionally, economic processes follow a linear model, starting with resources and ending in waste. This model has both ecological and economical sustainability challenges and leads to ever-increasing environmental pollution. In 2022 alone 2.2 billion tons of waste were generated in the European Union [Eurostat, 2022a], of which 30.2% were put in landfills, 14.2% backfilled, 6.8% incinerated, and 8.4% disposed otherwise [Eurostat, 2022b].

Waste can be reduced through mindful resource use, cleaner and more efficient processes, and establishing an economy based on more sustainable resources, e.g., by transitioning from fossil fuels to renewable energies. However, these efforts still do not solve the inherent issues of a linear economy, and merely lessen the size of the problem. Thus, it is imperative that we also transition towards a circular economy, where products that would be at the end of their lifespan are reused, repaired, or recycled rather than discarded [Geissdoerfer et al., 2017].

Although limiting waste is at the core of a circular economy, it is unrealistic to eliminate waste entirely. Many products cannot be reused directly but still contain valuable resources that must be recovered. Consequently, recycling plays an important role in this challenge by mitigating the environmental impact of waste and reducing the dependence on finite resources, e.g., by replacing natural gas with biogas [Starr et al., 2015]. Additionally, recycling helps lower greenhouse gas (GHG) emissions by decreasing the energy needed for production processes and resource extraction.

Recycling should be considered in every phase of a product's life cycle, from design to disposal. In this thesis, we focus on the phases after the waste ends up in the trash bin. Concretely, this starts when waste from various sources, including households and commercial establishments, is collected. Then the waste is sorted, separating recyclable materials from non-recyclables. Finally, the recyclables are processed, either being

transformed into raw materials for further manufacturing or, in some cases, directly into products like compost, which can be reintroduced into the circular economy. A highly important challenge in all phases of recycling is minimizing costs and increasing efficiency to make recycling economically viable.

Organic solid waste (OSW) includes solid materials with organic components generated in agricultural, industrial, and municipal settings [Guo et al., 2021]. For OSW recycling, the cost is mostly driven by harmful materials mixed into the waste, which need to be separated in a labor-intensive sorting phase before the organic materials can be composted and reused. Even worse, if the cost of waste separation is too high or separation is infeasible, the garbage is usually burned instead of recycled, increasing GHG emissions and losing valuable resources. In the waste collection process, a single heavily polluted stop can lead to the contamination of an entire truckload of recyclables. On the other hand, using several trucks to gather the waste separately increases both carbon emissions and costs. This illustrates the conflict between achieving effective pollution separation, managing expenses, and reducing emissions.

1.1 Problem Statement

The foundation of this work is existing data from a previous project by Heinzl et al. [2023] where mobile devices installed in OSW collection trucks captured images and enriched them with metadata such as time and location. The images were then analyzed with a neural network that identified harmful materials in the pictures with a confidence score. The severity of different pollutants in the waste is also given in advance using a scoring system. For instance, a piece of paper is assigned a low severity score, whereas a glass bottle has a higher score. Intuitively, the severity score represents the expected cost of removing the pollutant after collection. In the case of glass, it may break during handling, significantly increasing the cost of separation, while paper typically only needs to be removed when large quantities are found.

This raises the question of whether the efficiency and cost of OSW collection and sorting can be improved by optimizing waste collection tours based on the given historical pollution data, to separate clean and contaminated waste before it is mixed in the truck? More generally, can an initial data-driven sorting phase at the start of the recycling process reduce the total effort of sorting and collecting?

Two main benefits are expected from the aforementioned approach, which we refer to as *pre-collection sorting*. First, waste collected in clean routes can be recycled directly or with minimum labor, resulting in pure, uncontaminated compost. Second, post-collection trash separation efforts can focus on a smaller amount of truly polluted waste. Moreover, if the harmful materials are inseparable from the organic materials, a lesser amount of waste has to be burned or sent to landfills.

The task at hand can be broken down into two challenging subproblems: (1) predicting pollution of future organic waste collection stops. Based on the predictions, the second

subproblem can be addressed, which is (2) the optimization of OSW collection tours to optimize pollution separation, GHG emissions, and operational costs. Figure 1.1 illustrates a possible route schedule for a given day, with color-coded pollution data. Additionally, we aim to derive further insights from the data to guide future separation and collection efforts.

Existing solutions for predicting and managing pollution in waste management often use Internet Of Things (IoT) hardware to collect data directly in the bins [Toğaçar et al., 2020, Bakhshi and Ahmed, 2018, Kang et al., 2020], or focus on related but distinct tasks such as waste volume prediction [Kannangara et al., 2018, Rutqvist et al., 2019, Hannan et al., 2012] or outcomes of waste processing, such as biogas production and compost maturity [Wang et al., 2015, Xu et al., 2020]. While these approaches offer valuable insights, they cannot be directly applied to the task of predicting pollution of future collection events using only historical data.

Similarly, related works in vehicle routing problems (VRPs) and their variants [Caceres-Cruz et al., 2014, Wu et al., 2020] can serve as inspiration for the sub-task of optimizing waste collection tours, but require adaptations to fit the unique constraints of pollution separation.

1.2 Aim of the Thesis

This thesis is part of the Vienna Science and Technology Fund (WWTF) transfer project "Knowledge Graph-driven Tour Management for Sustainable Waste Processing" [1], which aims to apply knowledge and methodologies developed in the WWTF-funded project "Scalable Reasoning in Knowledge Graphs" [2] to optimize waste collection. As established in Section 1.1, there are two subproblems in this endeavor: (1) predicting pollution of future organic waste collection stops and (2) optimizing the tours to enhance pollution separation and reduce GHG emissions. Both subproblems require suitable knowledge representation. For that, we apply knowledge graphs, which are graph-based data models, enriched with (ontological) reasoning methods. They provide efficient organization and querying of the available data, with flexibility for further improvements.

An essential component in knowledge graph (KG) creation is the design of a suitable ontology, which serves as a structured representation of the domain. In this case, it provides a conceptual framework for modeling OSW collection. By incorporating a structured representation of knowledge specific to OSW, including the types and sources of waste and historical pollution data, we expect to make more accurate predictions of pollution levels. Moreover, the integration of this ontology with optimization algorithms allows for the design of efficient waste collection tours that separate clean and contaminated waste effectively and minimize GHG emissions. We express these aims in the following research question, which we address with a layered ontology presented in Chapter 4.

[1] https://www.wwtf.at/funding/programmes/ei/NXT22-018/
[2] https://www.wwtf.at/funding/programmes/vrg/VRG18-013/

Figure 1.1: Illustration of an (anonymized) route superimposed on Vienna.

Research Question 1. What is a suitable ontology for the domain of organic solid waste collection that supports pollution prediction and subsequent efficient tour optimization?

With a well-designed ontology in place, the next challenge is to apply optimization algorithms that use this structured knowledge as well as pollution predictions based on the available collected data from past routes. The goal is to not only enhance the separation of waste pollution in OSW collection tours but also to minimize their operational costs and environmental impact. More precisely, we have concrete functional and non-functional aims that the developed approach should satisfy:

1. The first objective is to maximize the amount of collected clean waste that is largely free of harmful materials such as plastics or glass. This leads to less burned waste and more usable compost. We want to accomplish this by assigning the waste collection locations to clean and polluted routes, the former containing historically more polluted locations. Multiple approaches for predicting pollution are explored, including ones that target specific harmful materials.

2. Secondly, optimizing only the amount of clean waste leads to longer routes, as clean stops may be further apart. Longer routes increase GHG emissions, which goes against economics and the greater goal of sustainability. In general, these objectives are conflicting. Which objective is more important depends on the context and should be tunable in a weighted objective function.

This leads to the following formulation of the second research question, which focuses on the practical application of optimization algorithms to improve outcomes in waste management.

> **Research Question 2**. How can optimization algorithms be designed and applied to waste collection tours to minimize operational costs and environmental impact while ensuring efficient separation of clean and contaminated waste?

In Chapter 5 we propose various methods for predicting pollution, including statistical measures, knowledge graph embedding models, and stochastic models to guide the optimization process. Then, in Chapter 6 we present a local search algorithm based on state of the art (SOTA) vehicle routing approaches, customized to the problem at hand.

The first two research questions naturally motivate the need to establish suitable metrics to ensure that the applied knowledge graph methods are improving waste collection. This requires determining how to measure the success of these applications effectively. This research question is addressed by formalizing the problem in Chapter 3 and evaluating in Chapter 7.

> **Research Question 3**. How can the effectiveness and efficiency of sustainable reasoning methodologies in waste collection optimization be evaluated, and what are appropriate metrics in this evaluation framework?

Additionally, the analysis of data enabled by the KG representation can reveal further insights, such as optimal depot placements or general suggestions for strategies for future waste separation and collection efforts. These insights contribute to a more comprehensive understanding of waste collection aspects, guiding further improvement of this waste management system.

1.3 Methodology

This thesis follows a structured approach that can be categorized as design science research. The methodology includes the following steps:

Literature Review A detailed review of existing research is done to identify SOTA methods in knowledge graphs, predictive modeling, and vehicle routing optimization, particularly in the waste collection domain. The review focused on understanding

limitations in existing approaches, which inspire the design of the proposed solutions of this thesis.

Problem Formalization and Ontology Design The concrete problem is formalized to create a clear framework for analyzing and solving the challenge of waste sorting before collection. A domain-specific ontology is designed to organize data for efficient reasoning and abstraction in a KG.

Predictive Models and Tour Optimization Algorithms We developed predictive models to forecast pollution levels at waste collection points using statistical techniques and machine learning methods. These predictions serve as input for tour optimization algorithms developed for the particular challenges and requirements of waste collection.

System Development We created a prototype system based on the proposed ontology, predictive models, and optimization algorithms. This system demonstrates how the presented solutions can be applied to real-world waste management and is used for the evaluation.

Evaluation and Validation The proposed methods were evaluated using metrics for prediction accuracy, route efficiency, and environmental impact. A real-world problem instance is used to validate the effectiveness of the solutions, showing their practical applicability.

1.4 Main Contributions

This thesis addresses an important challenge in waste collection and recycling by proposing methods to schedule routes such that potential pollution is separated in advance, thereby increasing the recycling rate and minimizing operational costs. A domain-specific ontology has been created to organize data into layers, supporting efficient reasoning by providing different degrees of abstraction. We research predictive models to improve pollution forecasting and explore performant methods to heuristically optimize waste collection routes. Metrics and methodologies are developed to evaluate and validate the proposed solutions through a prototype, providing practical and sustainable strategies for real-world waste management.

The main contributions of this thesis are:

- We introduce a formalization for the problem at hand, which we refer to as Pre-Collection Sorting Problem (PCSP). We introduce this formalization to provide a clear starting point for developing effective solutions and evaluation methods.
- We design a layered ontology for the waste collection and recycling domain, which organizes data into stages, enabling structured reasoning and efficient data manage-

ment. The ontology integrates data at different stages, including raw data, enriched data, and finally routing results.

- We propose algorithms for optimizing the PCSP. This entails optimizing waste collection tours, balancing objectives such as minimizing GHG emissions, operational costs, and pollution separation efficiency, and predicting future pollution based on historical data.

- We establish metrics and methodologies to assess the effectiveness of pollution prediction and tour optimization strategies. The applicability of the proposed algorithms is evaluated with a real-world problem instance.

- We integrate the developed models and ontologies in a prototype system, showcasing the feasibility and impact of the proposed solutions on real-world waste management challenges.

1.5 Limitations

Real-world problems typically come in a multitude of variants. For example, not every waste collection vehicle has the same GHG emissions for the same route, and not every meter of route length leads to the same increase in emissions. Accounting for all real-world complexities would overwhelm any practical attempt to reason about such systems. This is why reasonable assumptions and abstractions are essential, with this thesis being no exception.

We assume homogeneous vehicles that emit pollution proportional to the distance traveled and have a fixed waste capacity for a route. This abstracts away details such as changing traffic, road conditions, or additional emissions when the truck is fuller and, therefore, heavier. Each vehicle can only be assigned one route per day. Furthermore, we assume that a pickup stop always provides the same amount of waste. GHG emissions attributed to waste disposal are proportional to the volume of unrecycled waste and are independent of other details such as pollution, waste composition, or disposal method. We also assume that it is not possible to reschedule collection to a different day because clients are typically informed in advance when waste is collected.

Finally, we assume that the number of reports is always much greater than the number of pickup stops. And, we make the stronger assumption that this holds even if we only count reports with a unique location. This is necessary to simplify the thesis by eliminating edge cases of the problem that do not occur in practice.

Specifics on the assumed concrete values of constants in the prototype evaluation are discussed in Section 7.4.1.

1.6 Structure of the Work

The remaining thesis is structured as follows: Next, Chapter 2 gives an introduction to the theoretical background, including important concepts such as KG and related techniques. It provides a classification of VRP algorithms. After that, we provide a formalization of the problem at hand in Chapter 3. There we also explain which preprocessing steps are necessary to construct a defined problem instance for predicting pollution levels and routing waste collection tours. In the following Chapter 4 we focus on Research Question 1 and present the ontology specifically developed for this waste collection domain, and elaborate on the considerations for that particular design.

Research Question 2 is addressed in Chapters 5 and 6: The topic of Chapter 5 is pollution prediction, outlining the methods used to anticipate pollution levels for the subsequent waste collection routing. This leads directly to the already mentioned tour optimization, which we discuss in Chapter 6. As in the previous chapter, we explain which techniques are applied to the task and explain the thoughts behind our choices.

Then, in Chapter 7 the setup for the computational study part of this thesis is explained, and results are reviewed based on their strengths and weaknesses, thereby answering Research Question 3. Finally, we conclude in Chapter 8 with a summary of the findings and contributions of the thesis and give a final reflection on this work.

1.7 Declaration of Collaboration

Parts of this work were conducted collaboratively in a working group with Jonathan Lex. These include the design of the ontology (Chapter 4), the choice of statistical pollution prediction methods (Section 5.1), the development of the greedy tour optimization algorithm (Section 6.1) and the respective implementation in the prototype system. Our contributions were balanced, with most of the work in the mentioned parts being accomplished through co-working sessions.

CHAPTER 2

Preliminaries

This chapter presents fundamental concepts and literature forming the foundation of this thesis. We discuss clustering, ontologies, knowledge graphs, and associated methods relevant to structured data representation and reasoning. Furthermore, we explore solution approaches to the vehicle routing problem, including a section on local search, and specialized ontologies for the VRP. We also introduce existing approaches aiming to solve similar problems and their limitations concerning the problem at hand. In particular, we review machine learning (ML) literature on applications in waste collection, and present tour optimization literature. Throughout this thesis, we aim to complete this high-level literature introduction with separate research about particular techniques given in the corresponding sections.

2.1 Clustering Algorithms

Clustering is a well-known problem that is frequently used in ML, statistics and artificial intelligence. It involves dividing a collection of objects such that items within a group are more similar to one another, compared to items in different groups. In this context, these groups are referred to as clusters. There are many algorithms for clustering problems, including k-means [Ahmed et al., 2020], hierarchical clustering [Murtagh and Contreras, 2012] and DBSCAN [Schubert et al., 2017]. Also, KG techniques can be used to support clustering algorithms, in particular KGEs, because embedding data in a lower-dimensional vector space is a good basis for efficient clustering of complex data.

We evaluated several popular clustering algorithms and compared their strengths and weaknesses, the result of which is shown in Table 2.1.

Algorithm	Process	Advantages	Disadvantages
k-Means	Divides data into k clusters by minimizing the sum of distances between data points and their cluster centers.	Simple, fast, and efficient for large datasets; suitable for convex clusters.	Sensitive to initial centroids and outliers. Struggles with clusters of varying sizes and shapes.
Hierarchical	Builds a tree-like structure of clusters through an agglomerative (bottom-up) or divisive (top-down) approach.	Does not require a predefined number of clusters.	Computationally intensive for large datasets. Sensitive to noise and outliers.
DBSCAN	Density-based clustering groups points close to each other and marks points in low-density areas as noise.	Can find arbitrarily shaped clusters and handle noise. Does not require the number of clusters as input.	Struggles with varying density clusters. Sensitive to parameter selection.

Table 2.1: Comparison of Different Clustering Algorithms

2.2 Ontologies

Ontology, as a rather abstract notion, has been defined in various ways over time, with many similar interpretations [Guarino et al., 2009]. In this thesis, we use the definition of Studer et al. [1998], in which they describe an ontology as a specification of a shared conceptualization. In other words, an ontology is a collective understanding that allows different systems and users to interpret data consistently. Every conceptualization must have a certain scope or domain, which, in our case, belongs to the wider domains of waste collection and vehicle routing. Ontologies can be represented in various ways, including a whole hierarchy of specialized logics referred to as description logics [Baader et al., 2008]. Other notable representations are the OWL (Web Ontology Language) [Antoniou and Harmelen, 2009] and the graph-based RDF (Resource Description Framework) [Pan, 2009] that supports defining ontologies as graphs in text form via subject-predicate-object triples.

In this thesis, we borrow OWL terminology: For that, we define *classes*, *properties*, and *individuals*.

Definition 2.2.1 (Individual)**.** An individual in OWL represents a specific object or entity within the domain.

For example, there may be an individual pickup stop denoted as `Pickup Stop #1`.

Definition 2.2.2 (Class)**.** A class in OWL represents a set of individuals that share specific attributes or characteristics within a defined domain.

A class is a unary relationship, e.g., the individual `Pickup Stop #1` is in the class `isPickup`. That statement can be expressed as `isPickup(Pickup Stop #1)`.

Definition 2.2.3 (Property)**.** A property in OWL describes a relationship between pairs of individuals within a domain.

For an example, imagine the individual `Pickup Stop #1` is close to `Pickup Stop #2`. Then, there may be the property `isClose(Pickup Stop #1, Pickup Stop #2)`. Properties may have characteristics such as symmetry or transitivity. In the example, `isClose` is symmetric, meaning `isClose(Pickup Stop #2, Pickup Stop #1)` is latent knowledge, i.e., not explicitly stated, but inferrable.

Finally, we want to add attributes to properties. This is not directly supported in OWL. One option is to express this by replacing an attributed property with two properties and an additional class. For example, `property(A, B)` becomes `property1(A, X)` and `property2(X, B)`, such that the class `X` that represents the original property can have its own properties. This replacement is called *reification*. However, to reduce visual clutter, we use the OWL concept of *annotations* instead. While this has the drawback that annotations are usually ignored in OWL reasoners, we found it better suited to our reasoning techniques, as KGs typically support edge attributes.

Various tools exist for visualizing ontologies, with features designed for different use cases [Dudáš et al., 2018]. In this thesis, WebVOWL 1.1.7 [1] is employed because of features such as color coding and an intuitive drag-and-drop GUI. One shortcoming of WebVOWL is the lack of active development and resulting abundance of bugs, especially when loading ontologies from files. Also, annotations are not yet supported and will be addressed instead with remarks in the corresponding sections.

2.3 Knowledge Graphs

KGs are high-level, flexible tools from the research area of knowledge representation and reasoning. They support organizing and utilizing complex semi-structured information. As the name suggests, they are based on graphs, with nodes typically representing entities that can be either real-world objects or abstract concepts, and edges, which describe well-defined relations involving those entities [Ehrlinger and Wöß, 2016]. Typically, relations and entities have types with a real or abstract meaning. Often used are property graphs, where nodes and/or edges can have properties. Furthermore, KGs support (ontological) reasoning techniques, including sophisticated query answering. Unlike traditional databases, knowledge graphs provide reasoning for queries based on meaning rather than exact matches. Users can search based on types, entities and relations,

[1] https://github.com/VisualDataWeb/WebVOWL

allowing for more flexible retrieval of information. For a comprehensive introduction to knowledge graphs, we refer the reader to Janev et al. [2020], Ji et al. [2021]. Knowledge graphs are also frequently used where dynamic, ever-changing knowledge is modeled, and the flexibility and extensibility of KGs are required. For a semantic, knowledge-curating perspective on KGs see the work of Fensel et al. [2020].

In this thesis, we explore various KG reasoning methods, including logical knowledge and knowledge graph embeddings (KGE) [Bordes et al., 2013]. KGs also support graph neural networks (GNN) [Scarselli et al., 2008], which are special neural networks adapted to process graph-structured data. Additionally, KG reasoning methods include temporal knowledge graph embeddings (TKGE) [Leblay and Chekol, 2018], which are embedding models especially well suited to reason with temporal data. Furthermore, KGs allow for intuitive visualization of relationships and entities, which makes it easier for us to explore the data, identify patterns, and find deeper insights that might not be obvious in traditional data representation methods.

2.3.1 Logical Knowledge

Logical knowledge plays an important role in deriving insights from structured data in knowledge graphs. Using relationships and constraints defined within the KG, logical reasoning allows us to make latent information visible. For example, if a KG contains the facts “There is a book in the trash” and “Books consist of paper”, it can logically conclude that “There is paper in the trash” through transitive reasoning. This capability extends beyond simple relationships to include more complex logical rules based on ontologies, allowing the graph to validate data, check for inconsistencies, and enable complex queries. Another important application of logical knowledge is KG completion, that is inferring missing knowledge by applying logical rules to existing knowledge guided by the graph structure. Reasoning based on logical knowledge transforms a KG from a static collection of facts into a dynamic system that actively enriches data. Rule-based logical reasoning is used repeatedly in the prototype, implemented through the repeated application of Cypher [2] queries. The concept is exemplified by the Listing 2.1, where we perform the above inference example. We make this knowledge explicit by first matching the given scenario, and then constructing an edge from the *Report* node to the corresponding *PollutionType* node representing paper.

```
1  MATCH (r:Report)-[t1:tag]->(b:PollutionType), (p:PollutionType)
2  WHERE b.label="book" and p.label="paper"
3  CREATE (r)-[t2:tag]->(p)
4  SET t2.probability = t1.probability
```

Listing 2.1: Example of a cypher query

[2] https://neo4j.com/docs/cypher-manual/current/introduction

2.3.2 Knowledge Graph Embeddings

Knowledge graph embeddings, first introduced by Bordes et al. [2013], are a family of ML models that learn vector representations of entities E and relations R of knowledge graphs. By representing nodes and edges as low-dimensional vectors, embeddings try to encode the semantic and relational information from the graph in a more computationally manageable format. Then, based on these embeddings, KGEs use a scoring function $f : E \times R \times E \mapsto \mathbb{R}$ to compute the likelihood of a given subject-predicate-object triple (h, r, t) with respective embeddings (e_h, e_r, e_t).

For training, a loss is computed using positive triples $\mathcal{D}^+$ of the knowledge graph, as well as negative triples $\mathcal{D}^-$. These are usually generated under the local closed world assumption, which assumes that any triple that is not explicitly present in the knowledge graph is considered false. Importantly, this is limited to the context of the KG, i.e., to the entities and relations occurring in the graph. Finally, the vector representations are learned with the aim of maximizing the distance between the scores of positive and negative triples. This idea is formalized in Equation (2.1). This approach opens up new possibilities for tasks such as link prediction, where the embeddings can suggest potential connections between entities based on learned patterns. KGEs are also often used in clustering, recommendation, and similarity analysis, as embeddings can help detect nuanced relationships that may not be visible in raw data.

$$\mathbf{e} = \arg\max_{\mathbf{e}} \sum_{(h,r,t)\in\mathcal{D}^+} \sum_{(h',r,t')\in\mathcal{D}^-} f(\mathbf{e}'_h, \mathbf{e}_r, \mathbf{e}_{t'}) - f(\mathbf{e}_h, \mathbf{e}_r, \mathbf{e}_t) \tag{2.1}$$

In the following, we introduce three KGEs chosen to represent the hierarchy of expressivity and complexity in such models, which are then assessed in the task of predicting pollution levels in Chapters 5 and 7.

TransE Model

TransE [Bordes et al., 2013] is the foundational knowledge graph embedding model and is designed to embed entities and relations in a continuous vector space. The core principle of TransE is that relations between entities, i.e., edges in a KG, can be modeled as translations in the embedding space. For any triple (h, r, t), standing for the head, the relation, and the tail, TransE aims to embed them such that $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$. This means the vector representing the relation r can be understood as a translation from the head to the tail. Consequently, if we want to compute how likely an edge r' from h' to t' is, we check how close $\mathbf{e}_{h'} + \mathbf{e}_{t'}$ approximates $\mathbf{e}_{r'}$, e.g., using Euclidean Distance. A conceptual example of this can be seen in Figure 2.1. Due to its low dimensionality and simplicity, TransE is computationally efficient and works well for KGs with simpler relational patterns, such as one-to-one or many-to-one relations. However, there are some types of relations that TransE cannot represent well, such as symmetric relations, one-to-many and many-to-many relations. To give some intuition for the first case, a vector representing r cannot satisfy $\mathbf{e}_h + \mathbf{e}_r \approx \mathbf{e}_t$ and $\mathbf{e}_t + \mathbf{e}_r \approx \mathbf{e}_h$ without losing the

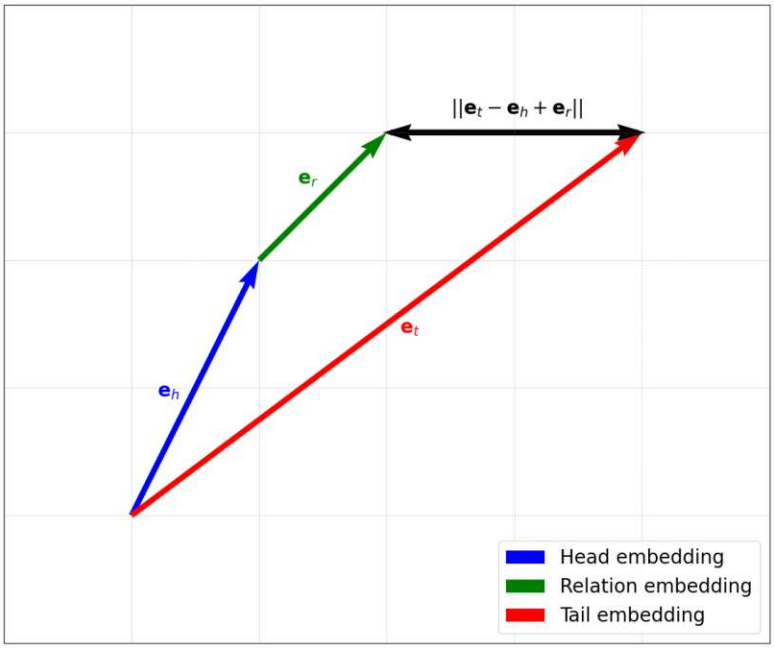

Figure 2.1: Illustrating the TransE embedding space in the plane.

distinction of h and t in embedding space. For the same reason, the same relation with the same head cannot result in multiple distinct tails, which would be required in embedding space for effective representation of one-to-many relations.

PairRE Model

PairRE [Chao et al., 2020] is a KGE model designed to address the limitations of earlier approaches like TransE. In contrast to models that use a single embedding to represent a relationship, PairRE computes two separate vectors for each relation: The first one scales the head entity embedding and the second scales the tail entity embedding. Given a triple (h, r, t), PairRE optimizes the embeddings such that $\mathbf{e}_h \circ \mathbf{e}_{r_1} \approx \mathbf{e}_t \circ \mathbf{e}_{r_2}$, where $\circ$ denotes element-wise multiplication, and $\mathbf{e}_{r_1}$ and $\mathbf{e}_{r_2}$ are the two relation-specific vectors. In other words, each relation maps to the tuple (r_1, r_2) in embedding space. A simple visual representation is shown in Figure 2.2. This dual-vector design enables PairRE to model complex relational types, including many-to-many, by capturing asymmetric and multiplicative interactions between entities. PairRE is still relatively efficient, even though it is more computationally expensive than TransE, and improves the ability to embed even more complex structures while maintaining scalability, making it a good compromise for large and heterogeneous knowledge graphs.

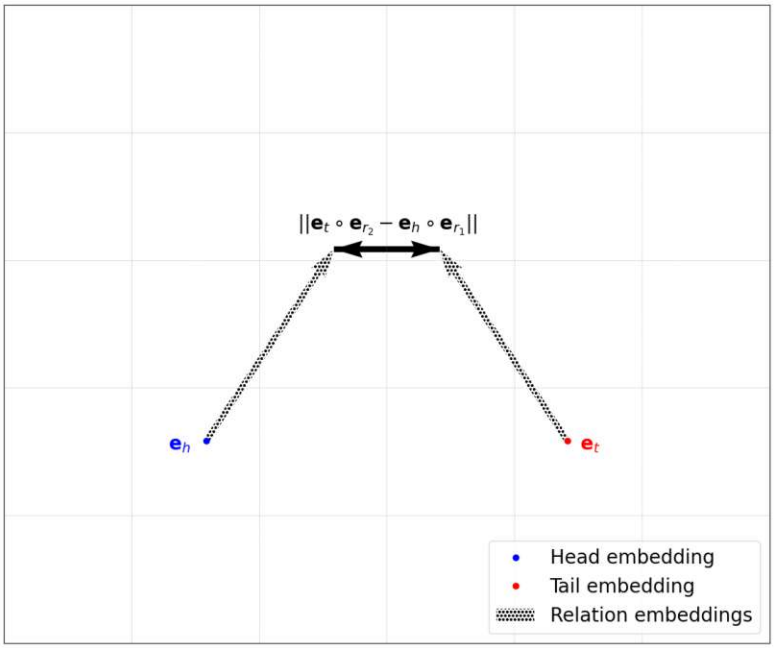

Figure 2.2: PairRE embedding space visualized in the plane.

TuckER Model

TuckER [Balažević et al., 2019] is a fully expressive KGE model, meaning it is able to capture any ground truth over entities and relations. It is based on tensor decomposition, specifically the three-mode Tucker decomposition [Tucker, 1966]. A knowledge graph is represented as a three-dimensional tensor, that is decomposed into three matrices and the core tensor. The entities and relations are embedded into the three matrices, which are the entity embedding matrix E twice, for subjects and objects in relations, and the relation embedding matrix W. Finally, the *core tensor* $\mathcal{W}$ captures interactions between the matrices. The tensor representing the entire KG is then $\mathcal{W} \times_1 E \times_2 W \times_3 E$, where $\times_n$ stands for the tensor product along the n-th mode. Given a triplet (h, r, t) with corresponding embedding vectors $\mathbf{e}_h, w_r, \mathbf{e}_t$, TuckER computes a score $\mathcal{W} \times_1 \mathbf{e}_h \times_2 w_r \times_3 \mathbf{e}_t$. that combines the embeddings of the head entity h, the relation r, and the tail entity t via the core tensor. This design enables TuckER to model complex and diverse relational patterns, including asymmetry, hierarchy, and transitivity. Using the Tucker decomposition, TuckER reduces the number of parameters compared to previous tensor factorization approaches such as ComplEx [Trouillon et al., 2016], improving the efficiency of such approaches consistently. However, while it is the most expressive KGE model that we explore, it is also the most computationally demanding.

2.4 Local Search

Local search algorithms modify an existing solution with different *move* operators aiming at iterative improvements by exploring the search space in proximity to the current solution. All solutions that can be derived from a solution S with a particular move operator x are collectively referred to as *neighborhood* $\mathcal{N}_x(S)$. There exist various strategies for choosing the next incumbent (solution), such as *first-improvement*, where the first generated solution found that is better is immediately chosen, or, *best-improvement*, where the whole neighborhood is generated and the best-found solution becomes the next incumbent.

Some strategies accept a worse solution with some small probability in order to escape local optima. One such method is simulated annealing [Kirkpatrick et al., 1983], which is inspired by the annealing process in metallurgy, where materials are heated and then slowly cooled to change their physical properties. This approach simulates the cooling process to first explore a solution space and escape local optima and then perform fine-grained optimization later in the process. This is done by accepting worse neighbors with decreasing probability when the system cools. At a set temperature, the probability of accepting a worse neighbor is proportional to the difference in objective. As shown in Equation (2.2), the temperature T starts at the initial temperature T_0 and decreases with each iteration i up to the maximum iteration i^{max}. The corresponding acceptance probability $\mathbb{P}_a$ for the incumbent solution S and proposed solution S' is stated in Equation (2.3).

$$T = T_0 \cdot \left(1 - \frac{i}{i^{max}}\right) \tag{2.2}$$

$$\mathbb{P}_a = max(1, e^{\frac{f(S)-f(S')}{T}}) \tag{2.3}$$

2.5 Vehicle Routing Problems

Vehicle routing problems (VRPs) represent a wide class of combinatorial optimization problems, which are concerned with finding the most efficient routes for a vehicle fleet to deliver goods or services to multiple locations while minimizing cost metrics like distance traveled or fuel consumption under real-world constraints. The distinction between delivery and collection lies solely in their meanings and does not influence decision-making for our purposes. Many variants of VRPs exist, reflecting the complexity of real-world vehicle routing. Similar to the setting in this thesis are the multi-depot vehicle routing problem (MDVRP), which is a generalization of the VRP where multiple depots are allowed, and the green vehicle routing problem (GVRP), which includes environmental concerns such as GHG emissions in the optimization process [Caceres-Cruz et al., 2014]. Depending on the variant, there may be maximum capacities assigned to vehicles [Ralphs et al., 2003], with either a homogeneous or heterogeneous fleet.

Furthermore, some variants generalize the problem by adding constraints such as time windows for deliveries [Kolen et al., 1987] or shift times for drivers [Ren et al., 2010].

VRPs are both complex (NP-completeness proven by Lenstra and Kan [1981]) and highly relevant for practical applications, especially in logistics and transportation. A variety of solution approaches have been proposed, which can be categorized as exact algorithms [Andelmin and Bartolini, 2017, Mingozzi et al., 2013], traditional domain-specific heuristics [Clarke and Wright, 1964], meta-heuristics [Azi et al., 2014, Baker and Ayechew, 2003, Bell and McMullen, 2004, Wu et al., 2020], and learning-based optimization, as discussed by Caceres-Cruz et al. [2014]. In the following paragraphs, we provide an introduction to each category.

Exact Algorithms. Exact algorithms like branch-and-bound and most integer programming approaches explore the solution space, usually with plausibility constraints, and can guarantee optimal solutions for small to moderately sized problems, but are usually worst-case intractable.

Domain-specific Algorithms. Domain-specific techniques are developed to exploit domain knowledge and include savings algorithms and insertion heuristics. They construct routes based on guiding principles, usually without backtracking, i.e., reversing decisions, do not guarantee optimality and tend to be more limited in their ability to explore the search space.

Meta-heuristics. Meta-heuristics include advanced techniques such as large neighborhood search [Azi et al., 2014], which iteratively destroys and repairs solutions to explore diverse neighborhoods, genetic algorithms [Baker and Ayechew, 2003], which evolve populations of solutions through selection, crossover, and mutation, and ant-colony optimizations [Bell and McMullen, 2004], where artificial ants build solutions based on pheromone trails that reflect the quality of paths in the explored routes. These methods are successful, because of their ability to explore large solution spaces and compute solutions within reasonable computation times.

An example of a meta-heuristic that offers SOTA performance for the VRP is the local search algorithm by Arnold and Sörensen [2019]. They use three complementary moves to explore a vast search space but also perform well-designed *heuristic pruning* of the neighborhood structures to avoid excessive computation. Furthermore, when the local search is stuck in a local optimum, they perform perturbation of the solution with a technique called *guided local search*. In particular, they penalize specific solution attributes based on observed characteristics of high-quality VRP solutions. They report solutions within a 0.25% range on more complex SOTA approaches on a broad VRP benchmark dataset, with equally short or even shorter running times.

Hybrid approaches. Another successful class of VRP algorithms are hybrid approaches, that combine two or more of the above techniques. An example of a hybrid

local search algorithm was proposed by Wu et al. [2020] to optimize a waste collection variant of the VRP that considers waste filling levels and prioritized waste. Their algorithm consists of an initial solution obtained from a particle swarm optimization, further optimized with a simulated annealing local search. Particle swarm optimization is a meta-heuristic, in which individual solutions, called particles, move through the search space to find optimal solutions by following their own best-known position and the best-known position of all solutions, i.e., the swarm.

Learning-based Optimization. A more recent development is the application of learning-based optimization techniques, promising advantages especially when domain knowledge is scarce or the problem is dynamic and complex. However, they often suffer from drawbacks introduced by the learning-component, such as limited interpretability, challenges when generalizing to unseen problem instances, and the additional demand for data and computational resources during model training. Learning-based techniques include enhancing meta-heuristics with machine learning or developing end-to-end learning models. For an overview of learning-based optimization and insights regarding advantages and disadvantages, we refer the reader to Li et al. [2022].

2.5.1 Vehicle routing ontologies

Some general optimization ontologies have been brought forward, for example, the well-known General Optimization Ontology (GOO) by Miller et al. [2004]. Furthermore, Agardi et al. [2022] proposed a general VRP ontology that can be adapted to concrete vehicle routing problems. Their model includes classes for vehicles, products, periods, values, and attributes, with subclasses for specific types of each. The model also includes classes such as travel time, distance, and reliability to support VRP optimization.

2.6 Traveling Salesman Problem

The Traveling Salesman Problem (TSP) is an optimization problem where a salesman must visit a set of cities exactly once and return to the starting point while minimizing travel distance. Unlike the VRP, which involves multiple vehicles and additional real-world constraints, TSP considers only a single route.

A simple yet effective intra-route move is `2-opt`, in which two edges in the route are selected and removed, and the segments are reconnected with two new edges to restore a valid route, as shown in Figure 2.3. This idea can be generalized to k-opt moves, where k edges are removed from a route, which is then repaired accordingly.

One of the most successful approaches for the symmetric TSP is the LK (Lin-Kernighan algorithm) [Lin and Kernighan, 1973], in particular the subsequent improvement by Helsgaun [2000]. The core idea is to explore k-opt moves for any k in a search tree that is based on observations about alternating paths and how these relate to k-opt. Essentially, the algorithm iteratively removes and reconnects up to k edges in a route to reduce the

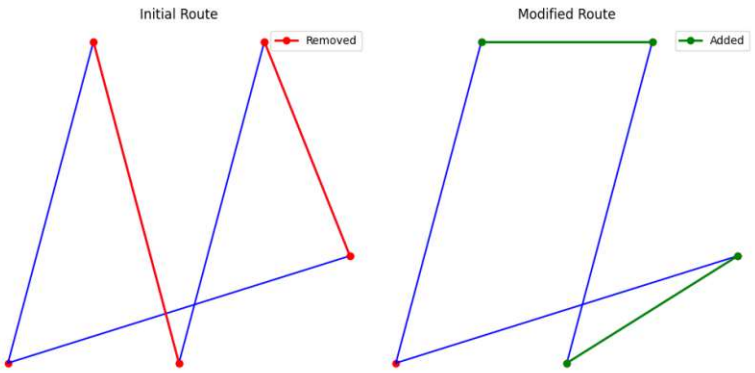

Figure 2.3: Example of a 2-opt move in an abstract route. The initial route is modified by removing and subsequently introducing edges s.t. the result is a valid route again.

overall cost. Unlike fixed k-opt methods, Lin-Kernighan dynamically changes which parts of the search tree are explored, enabling it to make deeper changes when necessary while avoiding excessive computation.

2.7 Tour Optimization Beyond Routing Algorithms

Related tour optimization literature comes primarily from the already mentioned domain of vehicle routing problems. However, additional insights outside the scope of VRP optimization may exist in other literature, such as the recommender system of Li et al. [2023], in which they apply a knowledge graph to cold-chain logistics. Their approach dynamically constructs a KG based on the output of a data mining module, which is then accessed by a recommendation module. They discuss the advantages of such a method over simply applying VRP heuristics, which can be summarized as a broader impact on the overall operation and a better adaptability to complex, uncertain problems.

2.8 Machine Learning in Waste Management

Literature exists for the related task of identifying pollution for waste classification before collection, but usually requires sensory IoT hardware in each bin [Toğaçar et al., 2020, Bakhshi and Ahmed, 2018, Kang et al., 2020]. Moreover, there are works focused on predicting waste volume either by land area [Kannangara et al., 2018] or individual bin [Rutqvist et al., 2019, Hannan et al., 2012]. These approaches cannot be directly applied to the problem of this thesis since we rely solely on historical pollution data. Furthermore, our focus is on predicting the purity of the waste instead of the volume. In this context, pollution prediction can be addressed in various machine learning (ML) tasks, such as binary classification (polluted, clean), regression, or stochastic modeling, to predict the expected severity or probability of pollution at a scheduled stop. It can also be addressed

as a link prediction problem in a knowledge graph: For example, if an edge $e = (p, t)$ is predicted, then at pickup p, we expect trash of the type t.

Other studies apply ML to predict the outcomes of processing particular waste, including compost maturity [Xu et al., 2020], pollution evolution [Alavi et al., 2019], biogas production [Wang et al., 2015], and dioxin emissions from incineration [Zhang et al., 2022]. These studies provide deep insights into waste composition and show suitable techniques for various predictions in that domain. However, they focus on different aspects of waste and as such can only serve as inspiration for predicting pollution before waste collection.

CHAPTER 3

Pre-Collection Sorting Problem

In this chapter, we define the task of separating pollution in waste collection routes and present a concrete mathematical formulation in Section 3.1. This serves to clarify any uncertainties from the initial descriptions and builds a foundation for designing specific solution methods.

Motivated by our aim for a direct, measurable impact, we aim to evaluate the results end-to-end, as close to the actual data as possible. However, the data recording process is limited and requires preprocessing for a meaningful evaluation, which will be elaborated on in Section 3.2. Furthermore, as real-world problems are usually highly detailed and complex, they benefit from abstraction and subsequent handling in several smaller tasks.

3.1 Formalization

In this section, the focus is on formalizing the main challenge of the thesis, that is, OSW pollution prediction and waste collection vehicle routing. The notation introduced in this section is used throughout the thesis. Supporting this, we provide a complete list of symbols in the appendix to aid the reader.

3.1.1 Problem Instance Specification

We refer to the problem at hand as Pre-Collection Sorting Problem (PCSP). A PCSP-instance includes a set of *waste collection reports* R, along with *locations* L and *timestamps* τ. The locations are given as geographic coordinates, i.e., latitude and longitude pairs. For any pair of locations, the distance function $\delta : L \times L \mapsto \mathbb{R}$ gives the travel distance from one location to another. [1] Also given is a list of *pollution types* T, along with the *severity function* $\sigma : T \mapsto \mathbb{R}$ that specifies how severe or detrimental the pollution is.

[1] Note that this is, in general, not symmetric, e.g., due to one-way roads.

Each report $r \in R$ has a location l_r, a timestamp τ_r, and a set of *tags* Θ_r, which can also be empty if no pollution was found or if the report is in the future. The set of all tags is defined as $\Theta = \cup_{r \in R} \Theta_r$. Furthermore, there are the functions $\gamma : \Theta \mapsto T$ returning the pollution type for each tag, and $\pi : \Theta \mapsto [0, 1]$ which maps any tag to a real number in the range $[0, 1]$. This value reflects how confident we are that the pollution was correctly identified, with higher values meaning greater confidence, and corresponds to the output of the neural network used for image recognition. Furthermore, we define the pollution of a report $\beta(r)$ in Definition 3.1.1.

Definition 3.1.1 (Pollution of a report)**.** The pollution $\beta(r)$ of a report r is defined as:

$$\beta(r) = \sum_{t \in \Theta_r} \pi(t) \cdot \sigma(\gamma(t)) \tag{3.1}$$

Intuitively, each waste collection report is a point in time in a tour where the mobile device in the waste collection truck records an image. However, it is important that reports are not confused with the actual waste collection stops in a tour. We refer to the logical entity that represents one or multiple bins emptied in a tour stop at a household or commercial building as *pickup stop* p. Since geo-coordinates across different days vary even if they are collecting the same bin, we try to clearly distinguish between the two notions. A pickup stop is, in general, scheduled repeatedly in various tours, but at most once per day. The set of pickup stops is then denoted as P. There are two special stops without reports, which are the *base station* b, where all routes start, and the *waste drop-off station* e, where all routes end. As per our assumption discussed in Section 1.5, the number of reports is much greater than the number of pickup stops $|R| \gg |P|$, even if we only count reports with a unique location $|R'| \gg |P|$, where $R' \subseteq R$, s.t. $\forall x, y \in R' : x \neq y \implies l_x \neq l_y$.

We expect a list of homogeneous *vehicles* V, each vehicle $v \in V$ having the same *capacity* C that is measured in the number of stops in a single tour that a vehicle can serve at once, i.e., the maximum number of stops $p \in P$ in a route minus two, because of b and e. We assume that each vehicle is able to perform one tour in a day and therefore assume $|V| \geq 2$ to allow for waste separation. Also, we assume that all vehicles have homogeneous travel distances given by the distance function δ.

Furthermore, there is a *clustering function* $\rho : R \mapsto P$ that assigns each report to a pickup stop. [2] The location l_p of a pickup stop p is then simply the average over all associated report locations.

To avoid overly complex notation, we denote the set of *days* as D, with every day being the collection of reports that correspond to that day $\forall d \in D : d = \{r \mid \tau_r \text{ is during day } d\}$. Moreover, relative to the *time of decision making* τ' we define the reports of the past as $R^{PAST} = \{r \in R \mid \tau_r < \tau'\}$. Furthermore, we denote the past reports associated with a given pickup stop p as R_p^{PAST}:

[2] Notice that ρ is always surjective since every pickup stop has at least one report.

Definition 3.1.2 (Past reports associated with a given report)**.**

$$R_p^{PAST} = \{r \in R^{PAST} \mid \rho(r) = p\} \tag{3.2}$$

Another useful definition is denoting the set of pickup stops, that are scheduled on day d with P_d, formalized as:

Definition 3.1.3 (Pickup stops in a day)**.**

$$P_d = \{p \in P \mid \exists r \in d : \rho(r) = p\} = \bigcup_{r \in d} \rho(r) \tag{3.3}$$

3.1.2 PCSP Solution Specification

In order to specify a PCSP *solution* S, we first have to formally define a *route*. A route, that is assigned to vehicle v on day d, is an ordered sequence $R_v^d = \{b, p_0, p_1, ..., p_k, e\}$, with $(p_i)_{0 \leq i \leq k} \in \{p \in P \mid \exists r \in d : \rho(r) = p\}$ being pickup stops that have a report scheduled on that day. A solution S is then given by a set of routes for each day $S_d = \{R_v^d \mid v \in V\}, S = (S_d)_{d \in D}$. S is *feasible* if for every day $d \in D$ the following properties are satisfied:

- Each pickup stop with a report scheduled in the day occurs in a route:

$$\forall S_d \in S : \cup_{R_v^d \in S_d} R_v^d \backslash \{b, e\} = \{p \in P \mid \exists r \in d : \rho(r) = p\} \tag{3.4}$$

- Any route may not exceed the given vehicle capacity:

$$\forall d \in D \; \forall v \in V : |R_v^d| - 2 < C \tag{3.5}$$

- No pickup stop occurs in two routes at the same time:

$$\forall d \in D \; \forall v_1, v_2 \in V : v_1 \neq v_2 \implies R_{v_1}^d \cap R_{v_2}^d = \emptyset \tag{3.6}$$

3.1.3 Objective Function

Before the objective function can be stated, a few intuitive notions have to be defined, starting with route length.

Definition 3.1.4 (Route length)**.** The *route length* $\lambda(R_v^d)$ of route $R_v^d = \{b, p_0, p_1, \ldots, e\}$ is the total distance traveled when visiting the location of the pickup stops in order:

$$\lambda(R_v^d) = \sum_{i=0}^{|R_v^d|-3} \delta(l_{p_i}, l_{p_{i+1}}) \tag{3.7}$$

We previously defined the pollution of a report in Definition 3.1.1. Next, we extend this concept to quantify the pollution of a waste collection event. Due to the data collection method, multiple reports can be associated with the same pickup stop on a given day. However, in reality, waste is collected at most once per day for a pickup stop. Therefore, a waste collection event is uniquely identified by a pickup stop and a day. To aggregate the pollution levels of individual reports within one such collection event, we use the arithmetic mean, which is efficiently computable and guarantees that all reports contribute equally.

Definition 3.1.5 (Pollution of a collection event)**.** The pollution $\beta(p, d)$ of a collection event, identified by the pickup stop p and day d, is defined as the average of the pollution of all reports for the pickup stop on the day d.

$$\beta(p, d) = \frac{\sum_{x \in \{r \in d \mid \rho(r) = p\}} \beta(x)}{|\{r \in d \mid \rho(r) = p\}|} \tag{3.8}$$

We can subsequently aggregate pollution from the collection events and finally define pollution in the context of routes. If the *pollution of a route* $\beta(R_v^d)$ is less than the threshold, i.e., $\beta(R_v^d) \leq \beta^{max}$ holds, then it is a *clean* route. Otherwise, the route is *polluted*. We considered both maximum and mean pollution across collection events for $\beta(R_v^d)$, but determined that the latter provides a more realistic model of route pollution, for the following reason: In the real world, pollution limits are typically specified relative to the total amount, rather than an absolute threshold, e.g., the world health organization recommends a limit of $10\mu g/l$ for lead in drinking water [WHO, 2022]. This also applies to the context of OSW recycling, even in extreme cases. For example, a single hazardous item, such as a car battery, can compromise an entire truckload. However, this does not contradict the use of a relative pollution metric but instead motivates choosing a sufficiently high severity score for such pollutants. Specifically, if a pollutant's severity exceeds $\beta^{max} * C$, any route containing that item is classified as polluted. The result of this choice is Definition 3.1.6. Following this definition of the route pollution, we additionally expect an *acceptance threshold* β^{max} from the PCSP-instance, which is the upper limit of the average pollution s.t. the route is classified as clean and can be recycled.

Definition 3.1.6 (Pollution of a route)**.** The pollution $\beta(R_v^d)$ of a route R_v^d is defined as the average of the pollution of all collection events in the route.

$$\beta(R_v^d) = \frac{\sum_{p \in R_v^d \setminus \{b,e\}} \beta(p, d)}{|R_v^d| - 2} \tag{3.9}$$

The objective function encapsulates all relevant environmental costs, which are, in our case, GHG emissions due to transportation, as well as the cost of unrecycled clean waste. Transportation cost is modeled as total route length times a cost coefficient $\epsilon_{collect}$, which can be interpreted as an approximation of the GHG emissions per distance driven by

the waste collection vehicle. Consequently, we specify this constant in grams of CO_2e (CO_2-equivalent) emissions per meter. Following, we anticipate that the main area of improvement from this optimization is a reduction of emissions from recycling instead of landfilling or burning OSW. For simplicity, we define a function ι that returns 0 on input R_v^d if the route is clean, and otherwise 1. In this thesis, we assume that polluted routes are not recycled at all. We reflect that assumption in the objective function and penalize unrecycled waste, i.e., all pickup stops that are scheduled to be picked up in polluted routes. Again, we multiply that number by a cost coefficient $\epsilon_{dispose}$, interpretable as the GHG emissions caused by not recycling but disposing of the waste from a pickup stop. An alternative interpretation, if the waste is recycled after all, is that $\epsilon_{dispose}$ encodes the additional post-collection separation effort needed to remove the pollutants from the recyclables. Either way, in this thesis we measure this quantity in grams of CO_2e per unrecycled tour stop.

$$f(S) = \sum_{S_d \in S} \sum_{R_v^d \in S_d} \lambda(R_v^d) \cdot \epsilon_{collect} + \iota(R_v^d) \cdot |R_v^d| \cdot \epsilon_{dispose} \tag{3.10}$$

Of course, since future pollution is unknown during the planning phase, $\iota(R_v^d)$ is also unknown in advance, presenting a key challenge that we address in Chapter 5.

3.2 From Data to Instances

Preprocessing, particularly for machine learning or artificial intelligence methods, usually consists of a few simple but important steps that prepare raw data for further analysis. In general, preprocessing aims to improve model accuracy, reduce training times, and help to prevent issues like overfitting. In this work, we checked for missing values, removed irrelevant information such as image duplicates, and, most importantly, analyzed the possibly faulty data to catch problems early on.

One notable finding occurred when initially plotting all reports on a map, as shown in Figure 3.1. We found reports from a suspiciously long tour spread over multiple days. Further analysis revealed a period where the truck was stationary and located at a truck repair shop, which also marked the furthest point from the depot in that route. We concluded that the truck had to go in for maintenance or repair work, and the mobile device mistakenly continued to record images. Thus, the data believed to belong to the tour, marked in blue, has been removed before further analysis.

3.2.1 Clustering Reports

In the context of this work preprocessing includes mapping reports to pickup stops, i.e., determining $\rho : R \mapsto P$.

This is a variant of the already mentioned problem called *clustering*, introduced in Section 2.1. Any criteria can be used for clustering, but in the case of the data at hand,

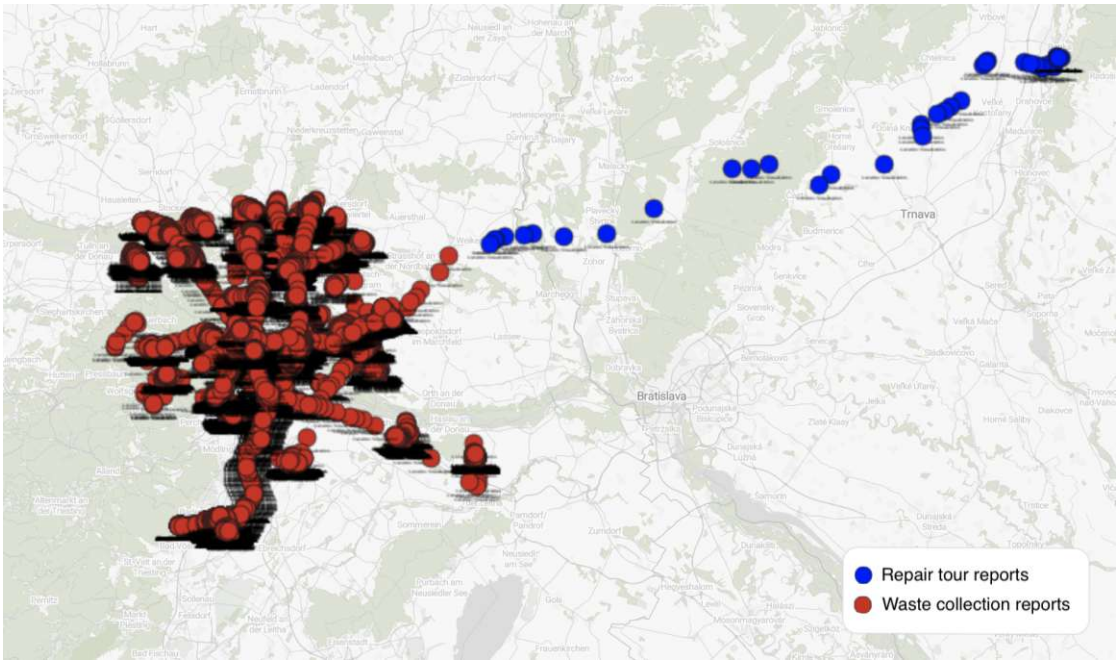

Figure 3.1: Map with all (anonymized) report locations. Reports marked in blue are not from a waste collection tour and are thus removed in preprocessing.

the most sensible choice is clustering by location only. This is because all other available information, such as pollution or time of waste collection, can change over time even for the same pickup stop.

Clustering by geo coordinates only is a rather low-dimensional variant of the clustering problem. Furthermore, the number of clusters is already given in the problem instance and corresponds to the size of the set of pickup stops $|P|$. This motivates applying k-means, due to the simplicity and our prior knowledge of the number of pickup stops. A potential disadvantage of this simple approach is that it has no knowledge on the concrete environment of a location. For example, two locations that are close may still be unlikely to come from the same stop, as they are separated by some obstacle such as a river. Furthermore, distinct clusters may be closer together in urban environments, compared to rural areas. Nevertheless, these are highly specific scenarios with a limited impact on the overall clustering accuracy.

Following, we extend Section 2.1 to give the reader a more detailed understanding of the chosen approach. The k-means algorithm groups a dataset into k groups or clusters, thus the name. Starting off, it randomly selects k points as cluster centroids. Each data point is then assigned to the cluster with the nearest centroid based on a chosen distance metric, in our case, Euclidean distance. After all points are assigned, the centroids are recalculated as the average of the points in each cluster. This process repeats until the centroids stabilize or a maximum number of iterations is reached.

3.2.2 Distances and Travel Times

The raw data does not contain distances, and travel times can only be partially observed in past waste collection tours. This is not sufficient for the PCSP-instance, as we rely on some measure of route length to plan new routes.

In order to enrich the data in that aspect, we set up a Open Source Routing Machine (OSRM) [3] instance. OSRM is a powerful open-source routing engine that allows us to query the distance and duration that occurs when traveling the shortest path for any two locations reachable on public roads. We chose to use route distances instead of route duration to measure the length of the route in the objective function, as the distance is less affected by traffic and similar dynamic changes. Additionally, this aligns with how emissions are typically specified, namely relative to the distance traveled rather than the time spent driving.

OSRM only provides an approximation of routes. As of the latest available version (5.24.0), it does not account for several real-world factors. First, OSRM does not support trucks as vehicle type, consequently it cannot account for truck-specific road closures or speed limits. Second, OSRM does not consider dynamic traffic conditions or construction information. While these constraints make OSRM unsuitable for real-world truck route planning, it is sufficient for the scope of this project, as it offers a solid foundation upon which algorithms can be built and tested.

[3] https://project-osrm.org/

CHAPTER 4

Ontology Design

In this chapter we design an ontology for the PCSP domain, that conceptualizes the relevant objects and relations, with the intent of adding structure to the data to create an easy-to-understand, efficient knowledge representation that enables efficient reasoning including queries, KGEs and routing algorithms. Further demands on the ontology are a good integration with machine learning models, and a high level of separation between input and output to avoid conflicts between different approaches.

Concretely, a layered ontology is used to store data in different stages of the process. More specifically, a layer each for raw and preprocessed data, and multiple layers for algorithm-specific results. This architecture allows us to view data at different granularities, providing a systematic abstraction for data interpretation and reasoning. Another advantage of a layered architecture is the simplified definition of procedures for the creation, update, and deletion of data because there are clear rules for references between layers. Higher layers may depend on lower ones, but never the other way around. That is, if some data changes on the lower layer, it is guaranteed that updates have to be propagated solely to the relevant individuals, classes, and properties in the higher layers.

The remainder of this chapter discusses the developed ontology layer by layer in order of increasing abstraction. That is also the order in which the data is processed and enriched. In each section, the classes and properties occurring in the respective layer are introduced. Inter-layer subject-predicate-object triples, i.e. relations between layers, are discussed in the section about the layer containing the subject entity.

4.1 Data Layer

The first layer, which we refer to as LAYER_1, contains the data cleaned as described in Section 3.2. In order to structure the concepts within our ontology, we introduce classes to group related entities and define their relationships. This should not be confused with classes in software engineering, where classes are used as blueprints for the creation of objects. We represent the contents of this layer using the superclass *LAYER_1*, with all other classes in this layer defined as its subclasses. Many notions from the PCSP-instance definition are present in the ontology for LAYER_1. Together, these classes and relationships allow storing and tracking waste collection events and contextual data, such as pollution. In the following, we present an overview of our model for this layer. A visual representation of the ontology can be found in Figure 4.1.

- The *LAYER_1* class serves as categorization identifier for the first layer, with all other classes in this layer being subclasses thereof. In order to reduce visual clutter, this subclass relation has been removed from the figure.
- The *Report* class is central to the ontology, representing a pollution collection event, i.e., a report in the PCSP-instance. Each report can have zero or more associated pollution tags annotated with a probability. Moreover, each report has exactly one *status* property linking to static information about the event and exactly one geographical location.
- The *PollutionType* class represents categories of pollutants and includes a severity attribute to measure pollution impact.
- The *Status* class describes the state of a report and contains a timestamp indicating when the report was recorded.
- The *Location* class provides geographical coordinates with attributes for latitude, longitude, and altitude.
- The *AbstractStop* class acts as a superclass for stops in a waste collection tour that are assigned exactly one location each.
- Subclasses of AbstractStop include *PickUp*, representing tour stops where waste is collected, the waste drop-off stations class *DropOff*, and the *Base* class, which expresses the starting points of waste collection tours, with the latter two containing exactly one individual each in our instance.

4.2 General Abstraction Layer

The general abstraction layer LAYER_2, encoded by the class *LAYER_2*, models the preprocessed data. This layer expands the ontology by introducing the concept of distance

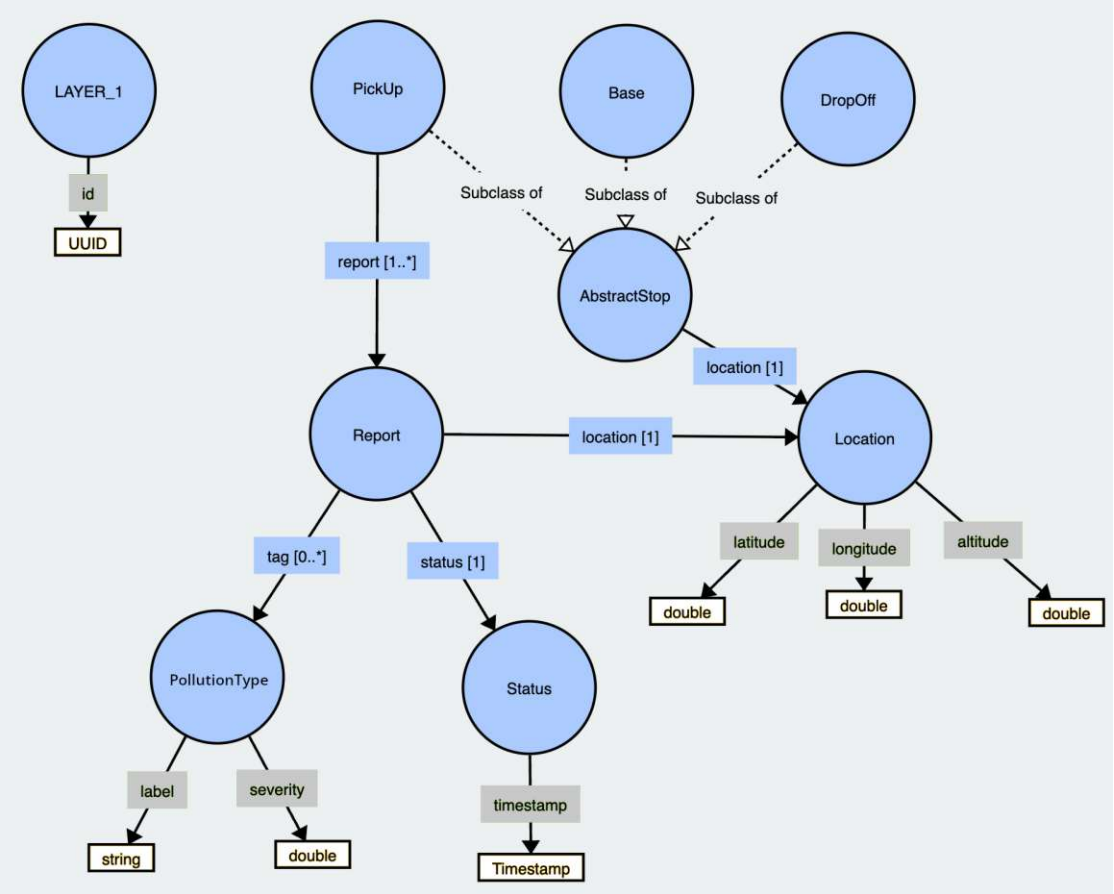

Figure 4.1: The Ontology for the data layer (`LAYER_1`).

between locations and improving how pickup stops are linked to their reports. It builds on the foundation of the first layer, integrates with it, and provides the structure to track and analyze spatial relationships.

For this, we perform clustering as described in Section 3.2 to group the `LAYER_1` *PickUp* nodes, and form clusters representing real-world waste collection location, such as households or industrial clients. The result is then expressed in `LAYER_2` *PickUp* nodes and the *contains* relation. We further enrich the distances between locations.

Classes, properties, and relations in this layer can be understood as abstraction or enrichment of the first layer. Instead of repeating the properties, we focus on the changes compared to `LAYER_1`. Furthermore, a visual representation of the ontology can be seen in Figure 4.2.

- At the foundation, *LAYER_2* is introduced as a categorization identifier for the second layer, with all other classes on this layer being subclasses. Again, this is omitted from the visualization.

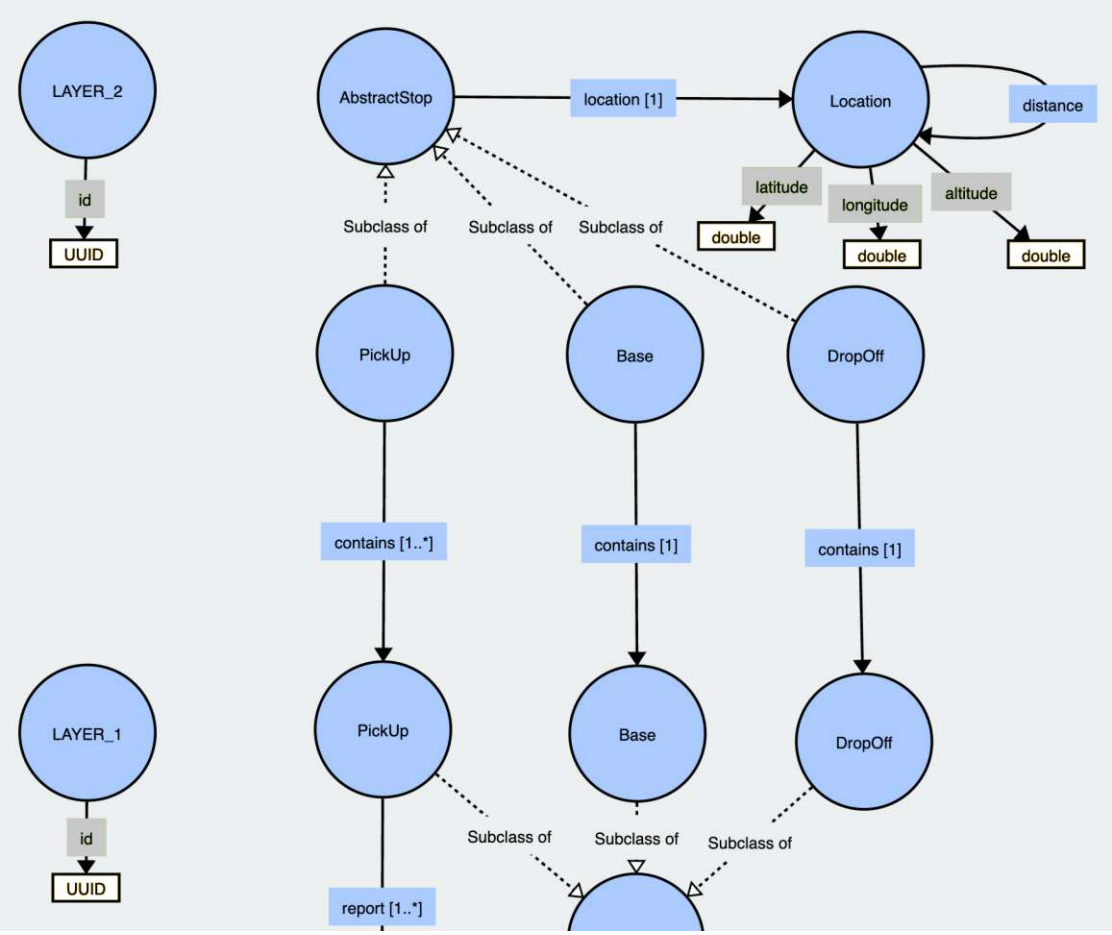

Figure 4.2: The Ontology for the general abstraction layer (LAYER_2).

- The *AbstractStop* class is used again on LAYER_2 as a superclass for different types of tour stops, which include *PickUp*, *Base*, and *DropOff*. In contrast to LAYER_1, these have only one property to signal which of their LAYER_1 counterparts they represent.

- Each *PickUp* contains one or more LAYER_1 *PickUp*s, while both *Base* and *DropOff* each contain exactly one instance of their respective lower-layer classes. This structure is chosen to allow for a higher degree of abstraction by grouping more pickup stops together.

- The distance and travel time between locations are modeled in the additional *distance* property of the *Location* class. They are annotated to the property in meters and seconds, respectively.

4.3 Routing Layers

The routing layers are collectively denoted as `LAYER_X` and individually referred to analogously by replacing the `X` in the name with the algorithm used to compute the waste collection routes. The routing layers build on previous layers, expanding their functionality to include route planning-related input and output. They extend the waste collection ontology by introducing classes related to routes, vehicles, and solutions. A visualization can be seen in Figure 4.3.

- As before, *LAYER_X* acts as a categorization class, with all other classes in this layer being subclasses of it. We make use of the property annotation to express that all properties, including *nextStop*, are in the corresponding routing layer as well.
- The *RouteInfo* class is introduced to represent waste collection routes. It includes a unique *routeId* and a *routeType* property to classify the routes as clean or polluted.
- A route is connected to a *SolutionInfo* entity that represents a PCSP-solution together with the associated routes. It encapsulates context on how the solution was derived, including the used metric for future pollution prediction, a threshold for clean pickup stops, and the emission coefficients used when computing the solution.
- Additionally, the ontology introduces the *Vehicle* class, which is linked to *SolutionInfo*. A solution can utilize two or more vehicles, each characterized by its capacity and emissions.
- Finally, the *AbstractStop* class of `LAYER_2` is further enriched with a *nextStop* property, enabling the representation of sequential stops in a route. The route to which the property belongs is annotated using an identifier from a *RouteInfo* individual.

4.4 Construction from a PCSP-instance

In this section, we discuss how to convert a PCSP-instance to fit the ontology, for example when constructing a KG.

For `LAYER_1`, the *Report*, *Status*, *PollutionType*, *Location*, *Base* and *DropOff* classes can be taken directly from corresponding data in the problem instance. In `LAYER_1`, a *PickUp* individual is created for each *Report* with a unique *Location*. If two or more *Reports* share a *Location*, then they share the *PickUp* individual.

The `LAYER_2` classes *Base*, *DropOff*, *AbstractStop* are straightforward to instantiate given the respective classes in the first layer. As explained in Section 3.2, clustering of the `LAYER_1` *PickUp* individuals is done using a k-means algorithm and expressed in

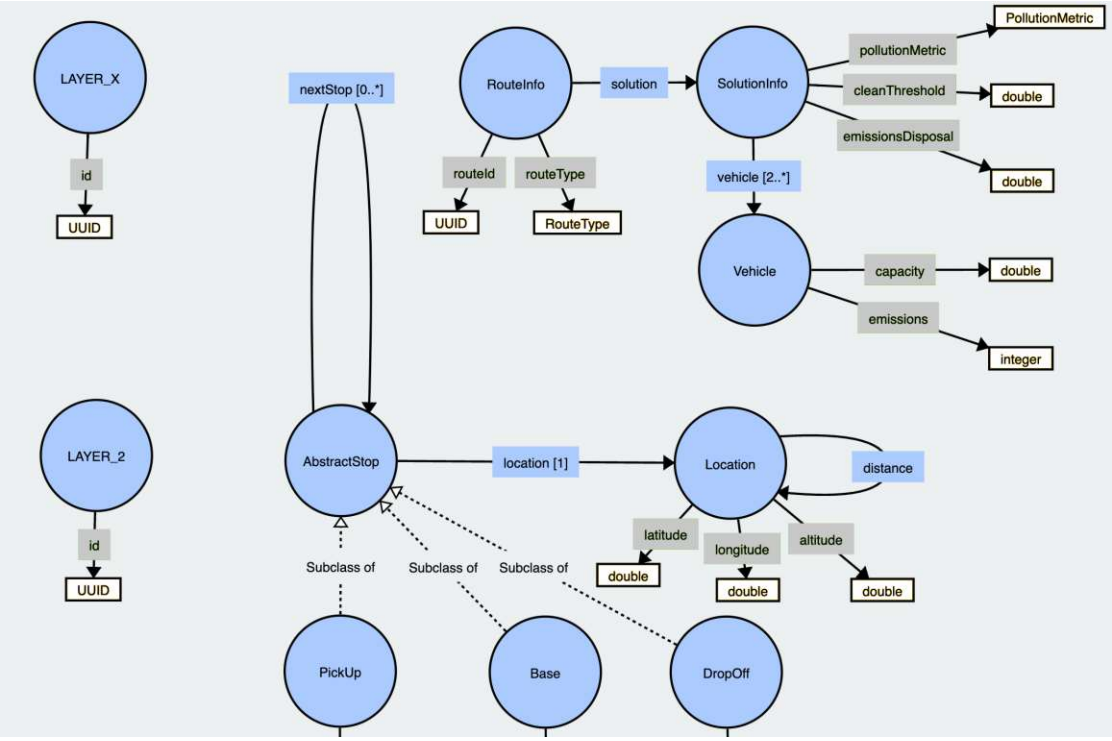

Figure 4.3: The Ontology for the routing layers (LAYER_X).

the problem instance as ρ function. Then, a total of $|P|$ new *PickUp* individuals are created, with *contains* properties according to ρ. The locations for the pickup stops are determined by averaging the coordinates of the clustered reports' locations. A drawback of this method is that the locations are not exact matches, but approximations of the real pickup stop locations, and are not necessarily positioned on the street network due to the averaging. Distances are expressed by adding a *distance* property to each tuple of *Location* individuals.

In LAYER_X, the *Vehicle* individuals, emission parameters, and pollution threshold are given in the problem instance, as V and C, $\epsilon_{collect}$ and $\epsilon_{dispose}$, and β^{max} respectively. Completing the layer requires a PCSP solution, determines the *RouteInfo* and *SolutionInfo* individuals. Additionally, *nextStop* edges are created to represent the sequence of pickup stops for each route.

CHAPTER 5

Pollution Prediction

In order to optimize future routes and increase the recycling quota, we need to predict which pickup stops are most likely to have pollutants and apply this insight to guide separation efforts during routing. More formally, when observing the objective function (Equation (3.10)) we find that the pollution status indicator function $\iota(R_v^d)$ is unknown in advance. Consequently, we require a strategy that allows us to perform the tour optimization regardless and indirectly optimize the objective. This is essential in solving the PCSP, because the quality of the solution strongly depends on the ability to create clean routes, and separate the polluted waste effectively.

This chapter explains two fundamentally different strategies to approximate $\iota(R_v^d)$. For both strategies, we require some form of pollution prediction, and for that we apply basic statistical measures, KGE models and probabilistic models. Approximations are consistently denoted with a hat symbol ($\hat{\cdot}$) over the exact variable.

5.1 Statistical and Latent Knowledge-Based Classification

Our first approach consists of predicting future reports' pollution using statistical measures and latent knowledge, then classifying reports and thus pickup stops as clean or polluted according to Definition 5.1.1. Finally, we apply this to approximate $\iota(R_v^d)$ by setting it to 0 if and only if all pickup stops in R_v^d are classified as clean.

Definition 5.1.1 (Clean and polluted stops)**.** All pickup stops on day d that have pollution below that threshold are regarded as *clean stops* $\mathcal{C}_d$.

$$\mathcal{C}_d = \{p \in P_d \mid \hat{\beta}(p,d) \leq \beta^{max}\} \subseteq P_d \tag{5.1}$$

The remaining stops are then referred to as *polluted stops* $\mathcal{D}_d$.

$$\mathcal{D}_d = P_d \backslash \mathcal{C}_d \tag{5.2}$$

5.1.1 Basic Statistic Measures

The simplest category of approaches discussed in this chapter is basic statistical measures. This group of statistics includes average, median, variance, and other point estimates, to summarize data without complex training or modeling procedures.

An intuitive idea is to apply the average of the pollution over the past reports as an approximation of future pollution.

$$\hat{\beta}_{avg}(p, d) = \frac{\sum_{r_i \in R_p^{PAST}} \beta(r_i)}{|R_p^{PAST}|} \tag{5.3}$$

One could argue that the median is a more robust choice. However, the average is much easier to compute than the median for large datasets. It requires only the sum of values and their count, letting databases process each value sequentially without storing them.

The average over all past reports means all reports contribute equally. However, when the reports are not distributed uniformly over the days, that metric results in a bias towards days with more reports. This motivates us to consider another option, which computes the average over the collection event pollution. If there is a constant number of reports per pickup stop for any given day, both metrics are equal. For that, we first need a definition of all days D_p that have at least one report associated with the pickup stop p.

$$D_p = \{d \in D \mid \exists r \in d : \rho(r) = p, d \text{ is before } \tau_r\} \tag{5.4}$$

This allows us to define the average collection event pollution, as explained previously.

$$\hat{\beta}_{avgday}(p, d) = \frac{\sum_{d_i \in D_p} \beta(p, d_i)}{|D_p|} \tag{5.5}$$

5.1.2 Knowledge Graph Embedding Models

Instead of approximating the collection event pollution $\beta(p, d)$ or report pollution $\beta(r)$, both being abstractions over the pollution tags, we propose a more ontological approach. Here, we predict the tags associated with the reports directly and then compute the above pollution metrics based on these predicted tags as per the definition of $\beta(r)$ and subsequently $\beta(p, d)$. There are two motivations for this approach. First, it plays to the strengths of KGEs, which learn from patterns within graph-structured data. Second, reducing pollution data to a single abstract value can obscure important patterns that may exist within the tag data. By targeting the tags directly, more of the structure remains and allows the model to identify patterns that may otherwise be missed.

Nevertheless, KGEs have limitations. For once, all relevant nodes must already be present during training, because KGEs cannot handle unseen nodes. Applied to our context, we

find that all future report nodes must be in the training data, meaning models require retraining whenever more reports are added. Also, future nodes do not have pollution, which could result in a bias in our model to associate reports further in the future with less pollution.

Furthermore, most KGE models are designed to work with the structure of the graph only, ignoring node and edge attributes, which may negatively impact prediction abilities if the attributes are relevant. Adaptations aiming to include such attributes have been proposed and applied with varying success: For node attributes, Kristiadi et al. [2019] propose adding a learnable parameterized function before the scoring function that takes the embedding along with the attributes of an entity and returns an adapted *joint embedding* vector. Pai and Costabello [2021] tackle the problem of incorporating a single, numerical edge attribute in the interval [0,1] for each edge by adding a so-called *FocusE* layer after the scoring function, and before the adapted loss function. This layer has the purpose of prioritizing triples with a higher edge attribute value. In contrast to the previous approach, this only influences the training process. In this thesis, we aim to use typical KGE models without such modifications but will encode some important information as triples. For example, we introduce nodes to represent the date in the timestamp property of the *Status* class, and add the corresponding relations. A disadvantage of this approach is that the encoding is not lossless, and information about the order of dates is lost.

Despite its limitations, TransE remains a popular and widely used baseline KGE model because it is simple and scales well, yet effectively captures basic relational structures. For those reasons, we include TransE in the evaluation as the baseline KGE approach. However, we want to predict the *tag* property, which is a many-to-many relation, and expect TransE to suffer from the aforementioned limitations in that regard. Consequently, we also apply the SOTA KGE models PairRE Chao et al. [2020] and TuckER Balažević et al. [2019], which complete a representative selection of the KGE hierarchy in terms of scalability and expressiveness.

5.2 Stochastic Modeling of Pollution

The second approach embraces the stochasticity of the problem and models pollution of reports of the same pickup stop p as samples from a stochastic process. In this framework, each pickup stop is treated as having its own, unique pollution distribution, independent of others, to model the variability and randomness in pollution levels specific to that location.

Then, in the tour optimization, we can compute the distribution of the route pollution based on the sum of the random variables associated with the pickup stops scheduled on that route, as stated in Definition 5.2.1. The summation of random variables corresponds to the convolution ($*$) of their probability density functions (PDFs), not to be confused with multiplication. Convolutions of arbitrary distributions do not have an analytical solution, with a few exceptions, notably including the Normal distribution. However,

using Theorem 5.2.1, we can approximate convolutions of any distribution. Applying the Fast Fourier transform has an asymptotic complexity of $\mathcal{O}$(n log n).

Theorem 5.2.1 (Convolution Theorem)**.** Let $\mathcal{F}$ denote the Fourier transform operator, $*$ the convolution operator and $f(x)$, $g(x)$ be two integrable functions. Then, multiplication in the frequency domain equals convolution of the original functions [Horváth, 2012].

$$f(x) * g(x) = \mathcal{F}^{-1}[\mathcal{F}(f(x)) \cdot \mathcal{F}(g(x)] \tag{5.6}$$

Definition 5.2.1 (Route Pollution Distribution)**.** Given a route $R_v^d = \{b, p_0, ..., p_k, e\}$ with k pickup stops and corresponding independently distributed random variables $(X_{p_i})_{0 \leq i \leq k}$ encoding the pollution distribution, then the route pollution is described by the following equation:

$$X_{R_v^d} \sim \sum_{0 \leq i \leq k} X_{p_i} \tag{5.7}$$

In other terms, let $f_{p_i}(x)$ be the PDF of random variable X_{p_i}. Then the function describing the PDF of $X_{R_v^d}$ is $f_{R_v^d}(x)$ and is defined by the following equation:

$$f_{R_v^d}(x) = f_{p_0}(x) * f_{p_1}(x) * \cdots * f_{p_k}(x) \tag{5.8}$$

Finally, the probability that the route pollution (Equation (3.1.6)) exceeds the threshold, i.e., $\beta(R_v^d) > \beta^{max}$, corresponds to computing a definite integral over the PDF, i.e., evaluating the cumulative density function, see Equation (5.9).

$$\hat{\iota}_{stoch}(R_v^d) = 1 - P(\beta(R_v^d) \leq \beta^{max}) = 1 - \int_{-\infty}^{\beta^{max} \cdot k} f_{R_v^d}(x)\ dx \tag{5.9}$$

In the objective function, we can replace $\iota(R_v^d)$ with this approximation, resolving the problem by explicitly modeling future pollution as a stochastic process. In the following sections, we explore four models for the pollution distribution of a pickup stop X_p.

5.2.1 Normal Distribution Model

An intuitive approach is to assume that pollution reports at a pickup stop follow a Normal distribution. The parameters of this distribution are the mean μ and standard deviation σ, which we estimate using the sample mean $\hat{\mu}_p$ and sample standard deviation $\hat{\sigma}_p$. To express the pickup stop for which we model the distribution, we use the subscript p. The formula for the sample mean is stated in Equation (5.10)), and for the standard deviation estimate, the Bessel corrected sample standard deviation is applied as per Equation (5.11).

$$\hat{\mu}_p = \frac{\sum_{r \in R_p^{PAST}} \beta(r)}{n} \tag{5.10}$$

$$\hat{\sigma}_p = \sqrt{\frac{1}{n-1} \sum_{r \in R_p^{PAST}} (\beta(r) - \hat{\mu}_p)^2} \tag{5.11}$$

A Normal distribution model has a drawback for our purpose: If there is not much pollution data for a pickup stop, then the results are overconfident because $\hat{\mu}_p, \hat{\sigma}_p$ are treated as if they are the true parameters. Empirically, at least 30 samples are required to get sensible estimates using the Normal distribution model, but for some pickup stops, there are much fewer past data points.

5.2.2 Student's t-Distribution Model

For small sample sizes, uncertainty is better captured by the Student's t-distribution rather than the Normal distribution. The t-distribution accounts for the degrees of freedom in the sample. The t-distribution is broader than the Normal distribution for small sample sizes, reflecting higher uncertainty in the estimate of the mean. As the sample size n increases, the t-distribution approaches the Normal distribution. We reuse the estimates $\hat{\mu}_p, \hat{\sigma}_p$ as defined for the Normal distribution model. Finally, the parameter referred to as *degrees of freedom* is $df = n - 1$, where $n = |R_p^{PAST}|$.

5.2.3 Bayesian Model

Bayesian models offer a structured way to perform statistical inference by combining prior knowledge with observed data to determine the posterior distributions of parameters. Using the Bayesian approach, we can explicitly include uncertainty in both $\hat{\mu}$ and $\hat{\sigma}$. Instead of treating them as fixed values, we assign prior distributions to these parameters (Equations 5.12,5.13) and compute the posterior probability distributions, which are finally sampled to derive parameters for the pollution distribution of the pickup stops' future collection events.

$$\hat{\mu} \sim \mathcal{N}(\mu_0, \sigma_0) \tag{5.12}$$

$$\hat{\sigma} \sim \mathit{HalfCauchy}(s_0) \tag{5.13}$$

For the prior parameter distributions, we choose a broadly parameterized Normal distribution for $\hat{\mu}_p$ (Equation (5.12)), and for $\hat{\sigma}_p$, we assume a Half-Cauchy distribution (Equation (5.13)), which is truncated at the center of the distribution. The effect of the truncation is that it has zero probability mass for negative values. Half-Cauchy was chosen over Half-Normal because it is less likely to over-constrain $\hat{\sigma}_p$, allowing the data to dominate the posterior when strong evidence is available. In other words, the Half-Cauchy distribution has more probability mass in the tail. For a visual comparison of both distributions, see Figure 5.1.

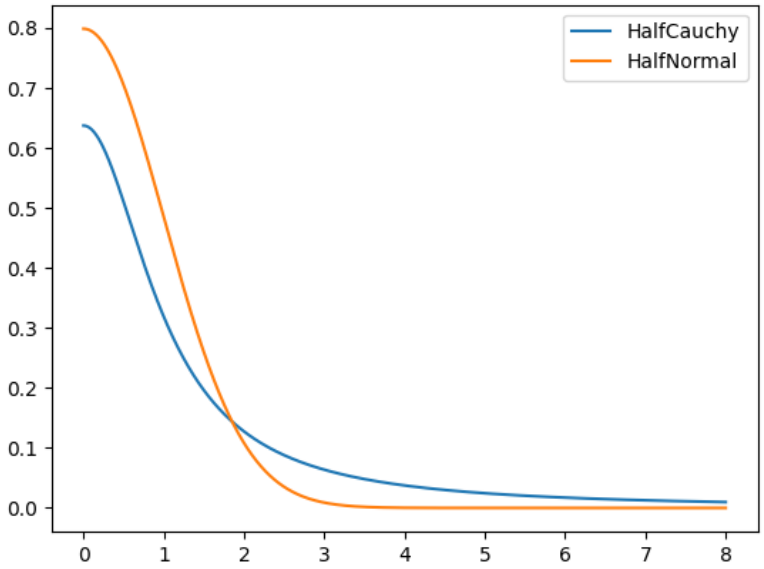

Figure 5.1: Comparison of Half-Normal and Half-Cauchy distribution

5.2.4 Bayesian Mixture Model

For our final model, we propose a Bayesian mixture of Normal distributions, motivated by real-world data that is often best described as a combination of multiple distributions. Concretely, we found that a weighted sum, i.e., a mixture, of three Normal distributions is a good fit for the observed pollution levels. The prior distribution over a histogram of actual pollution can be seen in Figure 5.2.

The Bayesian model consists of three Normal distributions, each with parameters μ, σ, and additionally the three weights $w_1 + w_2 + w_3 = 1$. In total, the posterior distribution of nine parameters is learned from the data. This complexity comes with a similar drawback as the simple Normal distribution, namely overfitting if there is not enough data. We conclude that each model has distinct theoretical advantages, and neither is strictly better in all scenarios. For empirical results in the context of this thesis see Section 7.6.2.

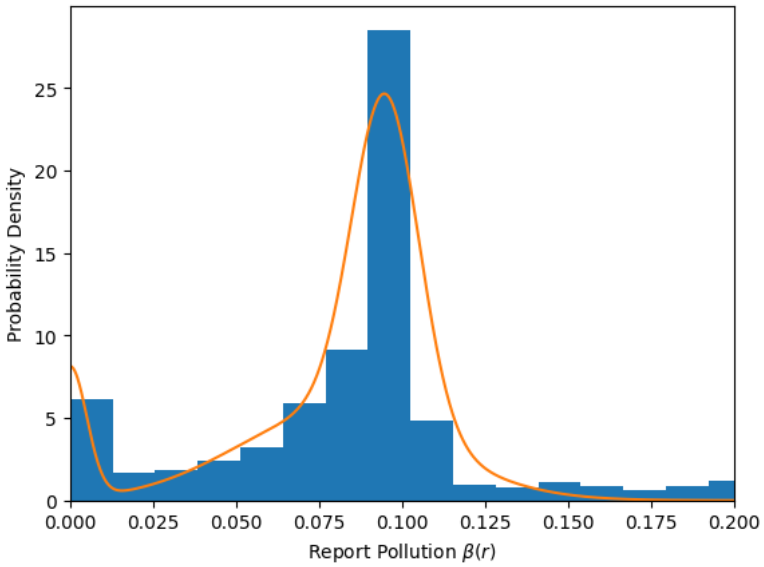

Figure 5.2: Histogram of report pollution (blue) and prior distribution of mixture model.

CHAPTER 6

Tour Optimization

Optimizing OSW collection routes under uncertain pollution levels is the key challenge that we address in this chapter. The proposed solutions aim to reduce costs and environmental impact, i.e., minimize the objective function outlined in Equation (3.10), indirectly, because of the unknown future pollution. We already characterized the routing subproblem as a vehicle routing problem. Depending on the strategy of pollution prediction, it can be approached deterministically or stochastically. Regardless of the strategy, we require a surrogate objective function $\hat{f} \approx f$.

In the first variant, we assume that the pickup stops can be categorized as (1) clean and (2) polluted, and then we schedule routes such that there are no polluted pickup stops in the clean routes. The opposite, i.e., a clean stop in a polluted route, is allowed and even preferred if it decreases the objective value overall. This is the case when the driven distance is reduced to an extent that the emissions from non-recycling are canceled out. Details on models for pollution classification can be found in Section 5.1.

The second variant deals with the stochastic nature differently by incorporating the confidence that we have of the route being clean in the objective function, as outlined in Section 5.2. For this strategy, any pickup stop can end up in any route as there are no dedicated clean routes. Instead, we directly optimize the expected objective value with no further constraints.

Due to the inherent complexity of VRPs, which are NP-complete even in their fully deterministic variants [Lenstra and Kan, 1981], we propose heuristic approaches. In this chapter, we design two PCSP tour optimization algorithms. First, we propose `GREEDY`, a simple heuristic that is intended as a computationally efficient alternative and baseline for a local search (`LS`) algorithm. For `LS`, we took inspiration from a SOTA approach for VRPs proposed by Arnold and Sörensen [2019], and adapted it to suit the objectives and constraints of the problem at hand.

In this chapter we repeatedly use the sequence concatenation operator, which we denote using the $\frown$ symbol.

6.1 Greedy Optimization

GREEDY is an algorithm that always performs the locally optimal step, i.e., it can be categorized as *greedy algorithm*, hence the name. This locally optimal step does not necessarily lead to a globally optimal solution, and much more sophisticated heuristics exist for VRPs. However, it offers two advantages: simplicity and computational efficiency.

Nearest Neighbor Routing. Our implementation is based on the nearest neighbor heuristic. We initialize a route, by starting at the base, and at each step, add the next pickup stop with the lowest distance to the previous stop. The concrete subroutine is shown in Algorithm 6.1. It expects as input a set of pickup stops, applies the nearest neighbor heuristic, and returns a valid set of routes serving the given stops. The validity of a route refers to the criteria defined in Section 3.1.2.

Algorithm 6.1: `Nearest-Neighbor-Routing`

Input: A set of pickup stops P, the base station b, the waste drop-off station e, the locations $L = (l_p)_{p \in P \cup \{b,e\}}$, the distance function δ, the vehicle capacity C

Output: A valid set of routes containing all $p \in P$

```
1  t ← 0;
2  r ← ∅;
3  Q ← P;
4  while Q ≠ ∅ do
5  |   r' ← (b);
6  |   i ← b;
7  |   while |r'| − 1 < C and Q ≠ ∅ do
8  |   |   q ← argmin_{q∈Q} δ(i, q);
9  |   |   r' ← r' ⌢ (q);
10 |   |   Q ← Q\{q};
11 |   |   i ← q;
12 |   end
13 |   r' ← r' ⌢ (e);
14 |   r ← r ∪ {r'};
15 end
16 return r;
```

Greedy Relocation. This could already serve as a valid initial solution, but there is a serious shortcoming that we want to address: It does not reflect the trade-off between route duration and pollution separation at all. We aim to account for this consideration

with a post-processing step shown in Algorithm 6.2. In it, each clean pickup stop is evaluated, and the additional distance caused by the additional stop in the clean route is compared to the distance of the optimal position in any of the polluted routes. If the emissions from the extra driven distance outweigh the emissions of one less recycled pickup stop, we move the stop to the polluted route.

Algorithm 6.2: `Greedy-Relocation-Optimization`

Input: A set of clean routes R_c, a set of polluted routes R_p, a set of clean pickup stops $\mathcal{C}_d$, the locations $L = (l_p)_{p \in P \cup \{b,e\}}$, the distance function δ, collection cost coefficient $\epsilon_{collect}$, disposal cost coefficient $\epsilon_{dispose}$, the vehicle capacity C

Output: A set of clean routes R'_c, a set of polluted routes R'_p

for $k \in \mathcal{C}_d$ **do**
  $r_c \leftarrow \{i \in R_c \mid k \in i\}$;
  $\delta_c \leftarrow d(r_c) - d(r_c \backslash \{k\})$;
  $g \leftarrow 0$;
  $m \leftarrow$ **null**;
  for $r_p \in R_p$ **do**
    if $|r_p| < C$ **then**
      $i \leftarrow \mathsf{argmin}_{1 \le j < |r_p|-1}\ \delta(l_{r_{p,j-1}}, l_k) + \delta(l_k, l_{r_{p,j}}) - \delta(l_{r_{p,j-1}}, l_{r_{p,j}})$;
      $r'_p \leftarrow (r'_{p,j})_{0 \le j < i-1} \frown (k) \frown (r'_{p,j})_{i \le j < |r_p|}$;
      $\delta_p \leftarrow d(r'_p) - d(r_p)$;
      if $g > (\delta_c + \delta_p) * \epsilon_{collect} + \epsilon_{dispose}$ **then**
        $g \leftarrow (\delta_c + \delta_p) * \epsilon_{collect} + \epsilon_{dispose}$;
        $m \leftarrow (r_p, r'_p)$;
      end
    end
  end
  if $m \neq$ **null then**
    $r_p, r'_p \leftarrow m$;
    $R_c \leftarrow R_c \backslash \{r_c\}$;
    $R_c \leftarrow R_c \cup (\{r_c\} \backslash \{k\})$;
    $R_p \leftarrow R_p \backslash \{r_p\}$;
    $R_p \leftarrow R_p \cup \{r'_p\}$;
  end
end
return R_c, R_p;

Finally, the complete `GREEDY` algorithm that integrates both solution components is shown in Algorithm 6.3. It produces valid solutions and is computationally efficient, but due to its simplicity, we anticipate far from optimal results. For instance, the Greedy-Relocation-Optimization method only ever reschedules one pickup stop at a time, yet

an improvement may only be obtained when relocating two or more consecutive pickup stops, i.e., a sub-route, at once. Furthermore, the nearest neighbor heuristic is clearly myopic because the impact on the rest of the route is not anticipated, and decisions are not reverted in greedy algorithms. We aim to improve on these shortcomings significantly with the LS heuristic.

Algorithm 6.3: GREEDY

Input: A set of clean pickup stops $\mathcal{C}_d$, a set of clean pickup stops $\mathcal{P}$, the base station b, the locations L, the drop-off station e, the distance function δ, collection cost coefficient $\epsilon_{collect}$, disposal cost coefficient $\epsilon_{dispose}$, the vehicle capacity C

Output: A set of clean routes R'_c, a set of polluted routes R'_p

1 $R_c \leftarrow$ `Nearest-Neighbor-Routing`$(\mathcal{C}_d, b, e, L, \delta, C)$;
2 $R_p \leftarrow$ `Nearest-Neighbor-Routing`$(\mathcal{P}, b, e, L, \delta, C)$;
3 $R'_c, R'_p \leftarrow$ `Greedy-Relocation-Optimization`$(R_c, R_p, \mathcal{C}_d, L, \delta, \epsilon_{collect}, \epsilon_{dispose}, C)$;
4 **return** R'_c, R'_p;

6.1.1 Complexity Analysis

Let $n = |\mathcal{C}_d|$ and $m = |\mathcal{P}|$. The first subroutine has quadratic runtime, $\mathcal{O}(n^2)$ and $\mathcal{O}(m^2)$ for the two calls respectively. That is because in each iteration of either of the `while` loops one pickup stop is scheduled, and in each iteration we check which remaining pickup stop is the closest, which runs in linear time because the check consists of linear constant time lookups. The second subroutine iterates over each clean pickup stop, and checks where among the $\mathcal{O}(n + m)$ stops scheduled in polluted routes the clean stop under consideration fits optimally, thus has a total runtime of $\mathcal{O}(n^2 + nm)$. Since constant factors are redundant in asymptotic bounds, the runtime complexity of GREEDY is simply $\mathcal{O}(n^2 + m^2 + n \cdot m)$.

6.2 Local Search Optimization

Almost all high-quality VRP algorithm designs are based on or at least incorporate meta-heuristics. They work based on the assumption that iteratively improving a solution leads to an overall good solution. Of course, counterexamples can be constructed in theory, but empirically local search techniques produce high-quality results at low computational cost. That is especially useful for combinatorial optimization problems such as the VRP, where exact algorithms are often computationally infeasible.

The LS algorithm used in this thesis is inspired by the local search method proposed by Arnold and Sörensen [2019] that was introduced in Section 2.5. We take inspiration from their choice of neighborhood structures, and adapt them to the problem at hand.

6.2.1 Intra-Route Optimization

The pickup-stop-to-route assignment remains the same during intra-route optimization, which means that only the distance penalty in the objective function is impacted. We are also only dealing with a single route. Consequently, performing intra-route optimization is essentially optimizing an Asymmetric Traveling Salesman Problem (ATSP).

Efficient ATSP Solvers. As discussed in Section 2.6, a highly effective approach to the TSP is the Lin-Kernighan algorithm [Lin and Kernighan, 1973]. Since the TSP is a heavily studied problem, there are already highly optimized implementations of LK. We apply LKH [1], which implements the Lin-Kernighan-Helsgaun algorithm, itself an improvement on LK, proposed by Helsgaun [2000]. This is one of the most successful solvers for the TSP, and frequently solves nontrivial instances with tens of thousands of cities to optimality. Furthermore, LKH supports ATSP by constructing a symmetric TSP instance by introducing dummy nodes.

From PCSP to ATSP. Mapping the PCSP intra-route optimization to a TSP instance is straightforward, except for one detail: Instead of a depot, where the tour starts and ends, we have a defined start and end node, which represent the base and waste drop-off location, respectively. This situation can be encoded by designating the base as depot and setting the distance from the drop-off location to the base to 0 and the distances from any other node to the base to ∞ (or sufficiently high value s.t. it is practically infinity). Any resulting tour must have the waste drop-off station as the second-to-last stop, which means we can just remove the very last return edge from the drop-off to the base and thus derive a valid PCSP route. Following these considerations, we parsed the problem to the TSPLIB format [2] expected by LKH.

6.2.2 Inter-Route Optimization

For inter-route optimization, we apply the CROSS-exchange (CE) operator [Badeau et al., 1997], which selects two route segments in distinct routes and then exchanges them. A simple example is shown in Figure 6.1.

In the context of the PCSP, CE requires additional checks to maintain the validity of the two routes, which additionally prunes the search tree. Following the classification strategy, pickup stops that are declared polluted cannot be moved to clean routes. This constraint is neither necessary nor justified when applying the stochastic strategy.

Pruning the Search Tree. In order to reduce the runtime of CE even further, Arnold and Sörensen [2019] *prune* the search tree, to focus on the segments that most likely offer improvements. Instead of considering all possible sub-routes, they only consider those that start with a *cross*, meaning that the sum of the distances to the first pickup

[1] http://webhotel4.ruc.dk/keld/research/LKH/
[2] http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp95.pdf

stop in each sequence must be shorter when swapped than in the initial routes. More precisely, assuming routes R_1^d, R_2^d, and p_1, p_1' as the stops that the sub-routes start with. Furthermore, let the initial edges be (p_0, p_1) and (p_0', p_1'), then it is a cross, if and only if the objective decreases with respect to the new edges $(p_0, p_1'), (p_0', p_1)$ and the reassignment of the starting nodes.

Applying this to the classification strategy, there are two cases that we consider:

- $\hat{\iota}(R_1^d) = \hat{\iota}(R_2^d)$:
  Both routes are clean or polluted. Clearly, the recycled volume does not change.
- $\hat{\iota}(R_1^d) \neq \hat{\iota}(R_2^d)$:
  W.l.o.g., let $\hat{\iota}(R_1^d) = 0$. Then, p_1 is clean as it could not be in a clean route otherwise. Likewise, p_1' must be clean, otherwise we could not reschedule it to R_1^d. Thus, the recycled volume does not change.

Therefore, following the classification strategy, it is enough to consider the distances when searching for a cross. Formally, the above situation is a cross exactly if:

$$\delta(l_{p_0}, l_{p_1}) + \delta(l_{p_0'}, l_{p_1'}) \geq \delta(l_{p_0}, l_{p_1'}) + \delta(l_{p_0'}, l_{p_1})$$

In the stochastic strategy, we cannot make this simplification. We have to evaluate the following inequality, which is essentially the objective function evaluated on the proposed swap of p_1 and p_1' while ignoring the unchanged sub-routes.

$$\begin{aligned}
&\delta(l_{p_0}, l_{p_1}) + \delta(l_{p_0'}, l_{p_1'}) \cdot \epsilon_{collect} \\
&\quad + (\hat{\iota}_{stoch}(R_1^d) \cdot |R_1^d| + \hat{\iota}_{stoch}(R_2^d) \cdot |R_2^d|) \cdot \epsilon_{dispose} \\
\geq \\
&\delta(l_{p_0}, l_{p_1'}) + \delta(l_{p_0'}, l_{p_1}) \cdot \epsilon_{collect} \\
&\quad + (\hat{\iota}_{stoch}(R_1^d \backslash \{p_1\} \cup \{p_1'\}) \cdot |R_1^d| + \hat{\iota}_{stoch}(R_2^d \backslash \{p_1'\} \cup \{p_1\}) \cdot |R_2^d|) \cdot \epsilon_{dispose}
\end{aligned}$$

After finding a valid start to a CE move, the next step is evaluating which sub-route lengths are optimal, and assess the total impact of the CE on the objective value. Since the previous simplifications do not apply, all possible lengths up to the end of the routes are assessed directly using the objective function. However, as this evaluation occurs only when a cross is found, we expect its effect on runtime to be limited.

Relocation Chain. Additionally to CE, Arnold and Sörensen [2019] apply the relocation chain operator for inter-route optimization because it affects more than two routes, unlike CE, extending the combined neighborhood size considerably. However, we observed that our data almost always contains only two routes because the capacity of two vehicles is sufficient. In the case of only two routes, CE includes the complete relocation chain neighborhood, and would therefore result in increased runtime with no expected improvements.

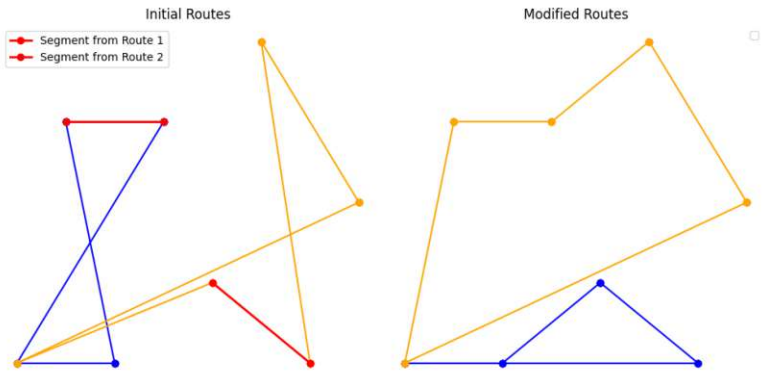

Figure 6.1: Example of a CROSS-exchange move. The route segments in red are swapped, potentially improving the solution.

6.2.3 Escaping Local Optima

Even with diverse and large neighborhood structures, getting stuck at local optima is generally not avoidable. There are different strategies for escaping local optima, including tabu search, destroy & repair and (knowledge-) guided local search. The latter is used by Arnold and Sörensen [2019], which we considered as well. However, knowledge guided local search does require domain-specific knowledge. In particular, more research is needed to determine if the characteristics of a good VRP route as observed by Arnold et al. translate to PCSP routes. They furthermore state that they observed less improvement from guided local search compared to the benefits of a more diverse set of neighborhood structures. Therefore, we decided to apply the simpler simulated annealing [Kirkpatrick et al., 1983] technique, because it is an established meta-approach for escaping local optima, that can be expected to work for the PCSP.

CHAPTER 7

Evaluation

In this chapter, we explain the evaluation procedure of the developed approaches, including an introduction of the dataset and the hardware used, and explain our choices for various coefficients and parameters. Following this, we present and analyze the results of the computational study performed on the prototype.

7.1 OSW Collection Dataset

We assess a real-world OSW collection dataset that was first introduced in Section 1.1 and then cleaned and enriched in the preprocessing step detailed in Section 3.2.

After preprocessing the dataset contains a total of 54 795 reports with 27 758 unique locations in both urban and rural areas over a period of five months, from January until May 2023. That is 148 days, including weekends. During preprocessing, we clustered the locations s.t. all reports were assigned to 2776 pickup stops, each with a unique location. Additionally, there are the waste drop-off location and the base station, both being located close to the center of all reports. As can be seen in Figure 7.1, the reports occur mostly from Monday through Friday, but occasionally, data was recorded on Saturdays and Sundays, too. With 2778 locations in total, there are 7 714 506 distinct-location tuples, for which we enriched distances in the preprocessing step, and subsequently saved them in the KG.

We take inspiration from the common 80/20 split and choose January through April as training data, with May remaining for testing. As the data is not perfectly uniform over the whole period, the actual split of the reports is closer to 75/25. An important statistic for the pollution prediction task is the number of past data points for each pickup stop that we need to schedule. As shown in Figure 7.2, many pickup stops have little to no previous collection history, which constitutes a key challenge in the pollution prediction task. This is also relevant because it helps to establish context for the interpretation of

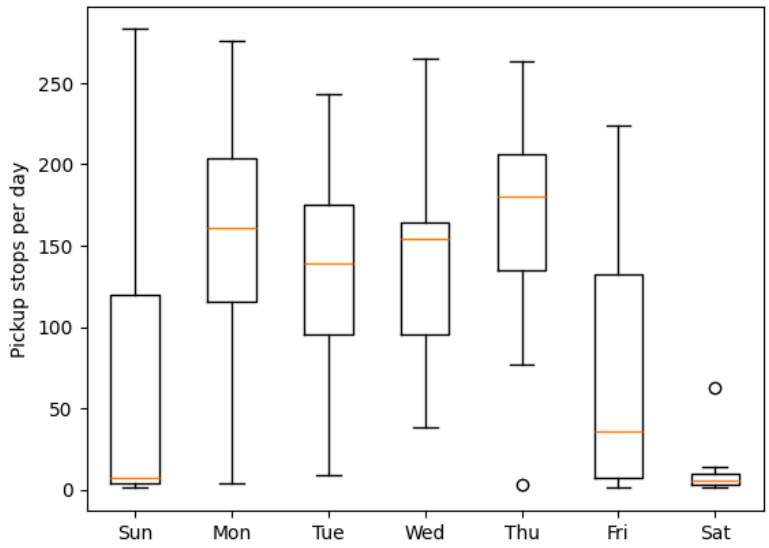

Figure 7.1: Statistics on the number of pickup stops by day of the week

the results. Notably, there are also some pickup stops that have a very high number of past collection events. This is unexpected, but we attribute it to artifacts of the data collection method. In particular, images were recorded even when the truck was not actively collecting waste. That means if the truck passes the same location many times on the way to other pickup stops, then reports pile up for that location.

Another possible challenge becomes obvious when observing the contribution of each pollution type to the total pollution per day (Figure 7.3). Apart from the clear importance of plastic foil, we observe high fluctuations in the total amount of pollution, even when accounting for weekly seasonal patterns. Furthermore, the respective significance of pollution types is less consistent than anticipated. For example, plastic bags contributed approximately a third of the pollution in January and the first two weeks of February, but only negligible amounts after that. Moreover, from 32 pollution types that we distinguish, 27 are observed only occasionally and do not significantly contribute to the total pollution.

7.2 Setup

The computational study was conducted in Docker [1] containers on a system equipped with the following specifications:

- CPUs: Intel(R) Xeon(R) Silver 4314 CPU @ 2.40GHz
- GPUs: Nvidia RTX A5000

[1] https://www.docker.com/

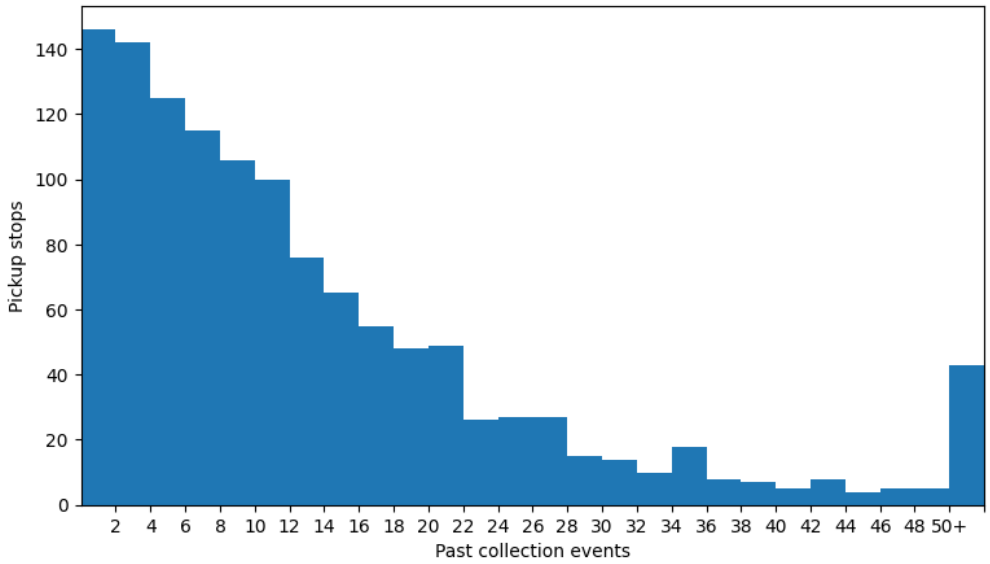

Figure 7.2: Distribution of the number of past collection events of all pickup stops being scheduled in May

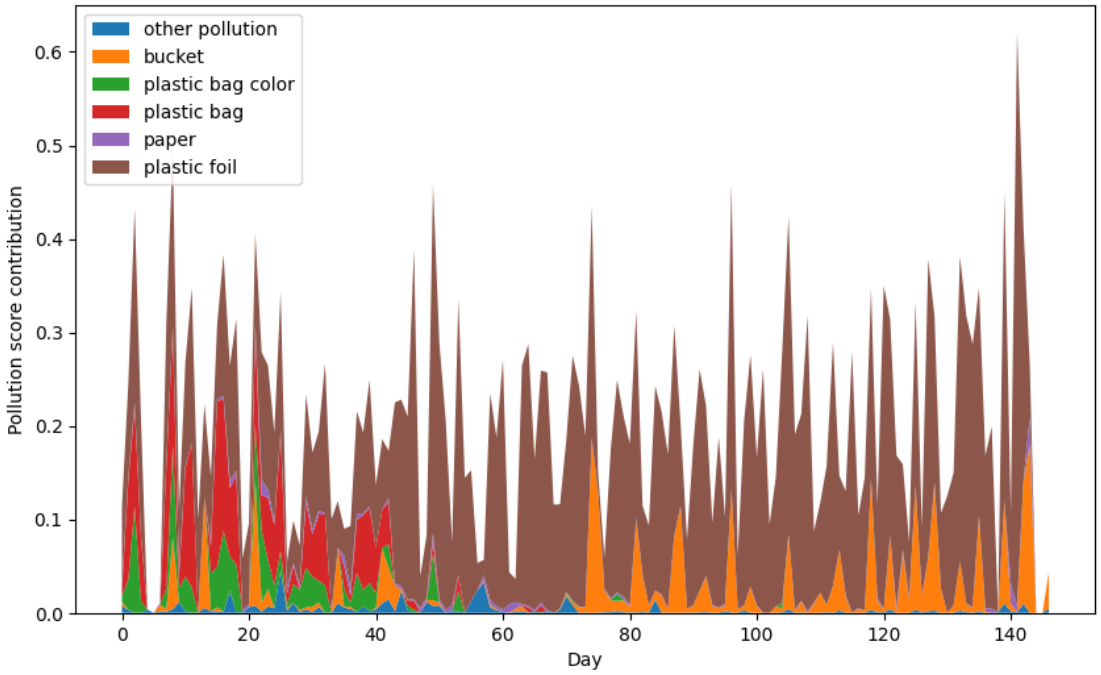

Figure 7.3: Absolute contribution of pollution types to total daily pollution. The total pollution for day d is $\sum_{p \in P} \beta(p, d)$, and the individual contribution is computed analogously with β accounting for only that pollution type.

- Memory: 1008 GiB
- Docker Version: 24.0.4
- Docker Image: node:16-alpine

7.3 Prototype

The prototype components have vastly different requirements, motivating a programming language-agnostic approach and used multiple programming languages. Memgraph [2] and its implementation of the query language Cypher was used to construct the knowledge graph based on the developed ontology. The KG was then populated with entities, relationships, and attributes from the given data made available in a MongoDB [3] database. The stochastic models were implemented in a Python application and for the Bayesian models, the probabilistic programming language Pyro [4] was used. For the KGEs, we used the package PyKEEN [5]. Finally, the routing algorithms were developed as a Node.js [6] application, written in TypeScript. Despite some performance trade-offs, this choice offered high-level development, good integration with supporting projects, and many libraries. We provide the code for the prototype in a Github repository [7].

7.4 Aims of the Evaluation

We evaluate the proposed system in three stages: Pollution Prediction, Tour Optimization, and their integration into the Pre-Collection Sorting Problem. First, the evaluation examines the ability of the pollution prediction models to accurately capture the patterns in pollution data to derive predictions. Next, it evaluates the effectiveness of the tour optimization algorithms by assuming that the predictions are correct. The objective of route optimization is to compute routes that balance emissions from transportation with those of waste landfilling or incineration. Finally, the evaluation looks at how well the framework integrates both pollution prediction and route optimization by observing the actual pollution caused by the scheduled routes.

7.4.1 Emission Coefficients

In the objective function (see Equation (3.10)) there are coefficients $\epsilon_{collect}$ and $\epsilon_{dispose}$. These can be understood as non-normalized weights whose ratio determine the importance of recycling in relation to the distance traveled. We could normalize the coefficients, since the absolute values are irrelevant for decision-making. However, we retain them

[2] https://memgraph.com
[3] https://www.mongodb.com
[4] https://pyro.ai/
[5] https://pykeen.readthedocs.io/en
[6] https://nodejs.org/en
[7] https://github.com/kglab-tuwien/waste-project

for interpretability, as they transform an abstract weighted sum into a GHG emission estimate measured in grams of CO_2e.

While our goal is to evaluate multiple ratios to demonstrate a broad spectrum of applications, we also want to highlight results based on realistic, real-world-inspired emission coefficients. To achieve this, we make the following assumptions:

Waste per pickup stop. We assume 240 liters of waste per pickup stop, which is the most common size for households in Austria. Arguably, this is too small for industrial clients. However, the assumption is reasonable in the context of this evaluation, because it is the most often used size and we assume homogeneous bins. The density of organic solid waste is around 0.1-0.4 kg/l depending on the composition and bin size. Larger waste volumes result in greater compression, as the weight of the accumulated waste compacts the material below. For example, Volkshochschulen [2023] estimates about 0.2 kg/l for 240 l OSW bins. In Lower Austria, Hannauer [2014] observed 0.15 kg/l on average in smaller cities and rural areas. Since this setting is close to the actual data, this is the density we assume. For 240 l bins, the expected waste is then 240 l $*$ 0.15 kg/l = 36 kg OSW per bin.

Disposal emissions. Lastly, we have to put a number on the emissions caused by polluted waste, whether that is generated by landfilling, burning the waste, or the emissions of separating the pollutant, if possible. Since there are many factors in this calculation, we have to make some assumptions. First, we assume that the waste is deposited in a landfill. It is quite difficult to put a number on the cost of separation because it depends strongly on the type and amount of pollutant, and published estimates on emissions caused by waste separation are hard to come by. A recent study about the GHG emissions of different means of processing by Nordahl et al. [2020] found that disposing via landfill is a very emission-intensive option with 400 gCO_2e/kg, and that is with a functioning gas capture system in place. Furthermore, they report that composting, i.e., recycling the organic waste led to negative emissions of -41 gCO_2e/kg, calculated based on credits for replacing conventional products that emit GHG such as inorganic fertilizers. This gives us a total estimate of 441 gCO_2e/kg that are emitted when the OSW is put in a landfill instead of recycled.

Transport emissions. To compare and evaluate tour distance and unrecycled waste we convert both to the expected GHG emissions measured in CO_2e. For the distance to emission calculation, we refer to a publication of the Austrian Umweltbundesamt [Austria, 2022], which puts the total emissions of a diesel truck under 18 tons at 608.7 gCO_2e/km. However, this includes any indirect emissions. For example, it includes replacing trucks after their expected lifetime, repair work, and maintenance. Since those are largely fixed costs independent of the distance driven, we use the direct emissions estimated to be 469.9 gCO_2e/km instead.

Putting it all together, our estimates for the emissions per bin are $\epsilon_{dispose} = 441*36 = 15\ 876$ gCO_2e/kg, the emissions emitted by transport for the collection are $\epsilon_{collect} = 0.4699$ gCO_2e/m and a realistic ratio of both is then $\frac{\epsilon_{dispose}}{\epsilon_{collect}} = 33\ 786$ m/bin. This also helps us with the interpretation, because for every additional bin that we expect to be able to recycle, we allow for an additional 33.8 km travel distance.

7.5 Procedure

In order to establish a baseline for the evaluation, we construct a status quo solution by extracting routes from the raw data. Since there is a timestamp and location for every tour stop, a route can be built by iteratively adding pickup stops to the route in chronological order. A necessary assumption due to the data collection method is that only the first occurrence of any pickup stop is relevant for the route construction. In the real data, there are multiple occurrences that we attribute to either inefficient routing or the need to drive past a previous pickup stop, without intent of collecting waste from that stop.

7.6 Pollution Prediction

In this section, we discuss the evaluation procedure and results of the pollution prediction task for both, the statistical and latent knowledge-based classification and stochastic modeling of pollution.

7.6.1 Statistical and Latent Knowledge-Based Classification

Following are the results from the first strategy for the pollution prediction task, in which we aim to classify future pickup collection events as either polluted or clean, as outlined in Section 5.1.

Hyper Parameter Optimization

The hyper-parameter optimization for the KGE models TransE, PairRE, and TuckER was done using PyKEEN's pipeline with early stopping enabled. We continued the optimization until the hyper-parameters converged and no improvement was observable for at least 100 iterations. About 1000 iterations turned out to be sufficient for all models, with TransE and PairRE converging rather quickly and TuckER showing improvement up to about 800 iterations. The resulting hyper-parameters can be found in Table 7.1.

Results

Even though the strategy is primarily about pollution classification, the prediction first computes the pollution estimate and then compares it to the threshold to derive a classification. This motivates a mean absolute error (MAE) and mean squared error (MSE) evaluation based on pollution predictions because it is more fine-grained than

Table 7.1: Hyper-parameters for PyKEEN models

	Hyper-parameter	**TransE**	**PairRE**	**TuckER**
Model	Embedding Dimension	80	256	224
	Relation Dimension	N/A	N/A	16
	Dropout	N/A	N/A	0.2, 0.0, 0.1
	Scoring Function Norm	1	N/A	N/A
	Vector Norm	N/A	1	N/A
Loss	Margin	2.815	9	N/A
	Adversarial Temperature	N/A	0.970	N/A
Optimizer	Learning Rate	0.00805	0.06566	0.0013
Negative Sampler	Ratio N/P	86	29	12
Training	Epochs	100	400	500
	Batch Size	4096	2048	219

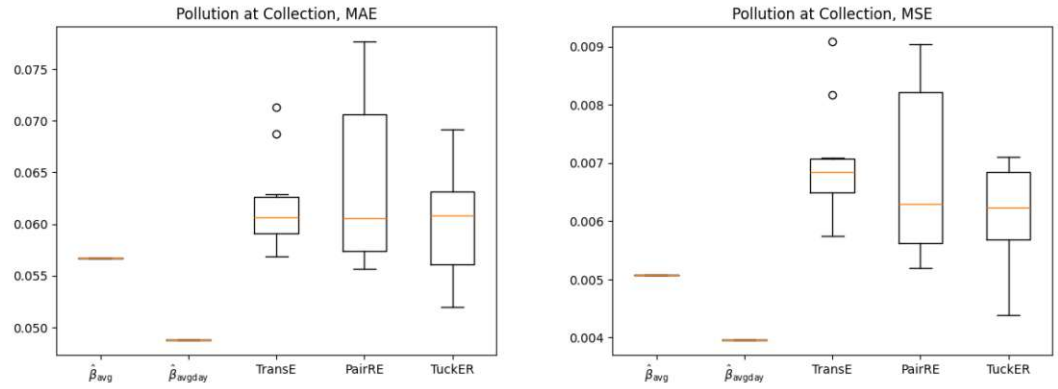

Figure 7.4: Prediction error (MAE, MSE) of statistical measures and KGE models

comparing true and false classification. Since KGEs are nondeterministic, we collect results over 10 runs and visualize the distribution of the respective results in a boxplot. Results from the KGE models and basic statistical measures are compared in Figure 7.4.

There are two unexpected outcomes: First, the results from the KGE models underperform, and have a higher error than the naive statistical measures in almost all runs. Second, the runs have much more variance than expected. Both outcomes could be explained by the KGE shortcomings detailed in Section 5.1.2. Another possible reason is that the structure of the ontology is not optimal for solving this specific task with KGEs, because edge and node properties are not embedded in the ontology structure. There is also a high amount of noise in the data, to the degree that even the best measure has a mean absolute error close to half of the pollution threshold itself.

The MSE and MAE show a similar picture, meaning the variance in error is comparable in all models. When comparing the average over past reports $\hat{\beta}_{avg}$ with the average over the daily mean $\hat{\beta}_{avgday}$, we observe that the latter has a lower error. Among the KGE

models, TuckER and PairRE have a slightly smaller median MSE compared to TransE, but due to the broad variance in runs, no model consistently outperforms the others. Overall, $\hat{\beta}_{avgday}$ has the lowest MAE and MSE, is deterministic, is efficiently computable, and is therefore our prediction model of choice for the remaining evaluation.

7.6.2 Stochastic Modeling of Pollution

This section presents the results of the second strategy for the pollution prediction task. This approach models the pollution of future pickup collection events as stochastic processes specific to each pickup stop, as detailed in Section 5.2.

An intuitive evaluation metric for a distribution and a set of observations is the likelihood. It computes the joint probability of the observations given the parameterized distribution. For continuous distributions, the PDF is evaluated, thus returning probability density in the interval $[0, \infty)$.

In this setting, we have a distribution per pickup stop, each with a separate set of observations, i.e., observed pollution in the test data. We evaluate two metrics, that combine the individual likelihoods differently: (1) the Average Likelihood computes the average over all observations, and (2) the Normalized Likelihood, which gives equal weight to each distribution, regardless of the number of observations. The formulas for these metrics are in Equations (7.1) and (7.2).

$$\text{Average Likelihood} = \frac{1}{N} \sum_{p \in P'} \sum_{i=1}^{n_p} f_p(\beta(p, d_i) \mid \Theta_p) \tag{7.1}$$

$$\text{Normalized Likelihood} = \frac{1}{|P'|} \sum_{j=1}^{P'} \frac{1}{n_p} \sum_{i=1}^{n_p} f_p(\beta(p, d_i) \mid \Theta_p) \tag{7.2}$$

where:

- P' are the pickup stops scheduled in the future
- n_p is the number of data points for the pickup stop p
- $f_p(\beta(p, d) \mid \Theta_p)$ is the probability density function (PDF) of the distribution associated with a pickup stop p, evaluated on the observed pollution $\beta(p, d)$
- Θ_p are the parameters of the distribution
- $N = \sum_{p \in P'} n_p$ is the total number of observations across all distributions

The results can be seen in Table 7.2. In both metrics, the Student's t-distribution and Bayesian mixture model outperform the normal distributions in Bayesian and classical statistics variants. Specifically, the Student's t-distribution achieves the highest

Table 7.2: Stochastic pollution models: Average and Normalized Likelihood, higher is better

	Normal	Student's t	Bayesian Normal	Bayesian Mixture
Average Likelihood	0.983	**3.013**	2.054	2.795
Normalized Likelihood	1.051	**1.791**	1.627	1.727

average likelihood, indicating its superior ability to fit the data when considering the overall likelihood across all observations. Similarly, the Bayesian mixture model shows competitive performance, particularly in the normalized likelihood metric. The classical normal distribution performs by far the worst, suggesting that our other models effectively mitigate the overconfidence that we predicted for the normal distribution. However, there is a clear cost to both the Student's t-distribution and the Bayesian mixture model, namely that no analytic solution exists for convolutions. As described in Section 5.2, there are methods for computing convolutions even if there is no analytic solution, but that necessarily increases the runtime.

7.7 Tour Optimization Results

In Table 7.3, we evaluate the tour optimization task with different values for the emission coefficient ratio between 0 and $1.00e6$ to investigate a broad range of emission scenarios, with $3.38e4$ corresponding to the realistic scenario laid out in Section 7.4.1. For each ratio, we consider four solutions: The first is the status quo solution extracted from the real-world data, serving as a baseline. Next, we compare results from the `GREEDY` algorithm that optimizes the objective function with a classification strategy based on the pollution approximation $\hat{\beta}_{avgday}$. The third and fourth solutions are derived using the `LS` algorithm that iteratively improves the `GREEDY` solution. The third and fourth solutions differ in the objective function, with the former using the same classification approach, while the latter applying the stochastic strategy based on the Bayesian normal distribution model. The results are ambiguous for that same reason. The stochastic local search solutions consistently produce the shortest routes. The predicted disposed bins are also far lower than for both solutions based on the classification strategy, but that can be attributed to the different pollution prediction technique. Overall, the status quo is by far the worst in all scenarios, in both the number of disposed bins and the driven distance. Generally speaking, the higher the ratio of emission coefficients, the more focus is on recycling over the total route length, which is also reflected in the results.

7.8 PCSP Results

Finally, we assess the PCSP end-to-end by computing the actual pollution of the routes, with which we get $\iota(R_v^d)$ and evaluate the objective function $f(S)$ of solutions introduced in the last section. This effectively measures how well the prediction strategy and tour

Table 7.3: VRP results: Summary of recycling efficiency and sustainability metrics based on predicted pollution

Algorithm	Strategy	$\frac{\epsilon_{dispose}}{\epsilon_{collect}}$	Predicted Emissions [CO2e]			Distance	Disposal	Runtime
			$\hat{f}$	Collection	Disposal	[m]	[stops]	[min]
A STATUS QUO		0	6.90e+9	6.90e+9	**0.00e+0**	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	0	1.46e+9	1.46e+9	**0.00e+0**	3.10e+6	3971	15.79
LS	Class., $\hat{\beta}_{\text{avgday}}$	0	**1.24e+9**	**1.24e+9**	**0.00e+0**	**2.64e+6**	3971	63.15
LS	Stoch., B. Normal Dist.	0	1.47e+9	1.47e+9	**0.00e+0**	3.14e+6	**795**	151.92
A STATUS QUO		250	8.76e+9	6.90e+9	1.86e+9	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	250	2.56e+9	2.36e+9	1.94e+8	5.03e+6	1652	15.94
LS	Class., $\hat{\beta}_{\text{avgday}}$	250	2.31e+9	2.11e+9	1.95e+8	4.50e+6	1663	46.80
LS	Stoch., B. Normal Dist.	250	**1.45e+9**	**1.37e+9**	**7.45e+7**	**2.92e+6**	**634**	218.82
A STATUS QUO		500	1.06e+10	6.90e+9	3.73e+9	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	500	2.74e+9	2.39e+9	3.54e+8	5.08e+6	1507	15.93
LS	Class., $\hat{\beta}_{\text{avgday}}$	500	2.49e+9	2.13e+9	3.55e+8	4.54e+6	1513	50.60
LS	Stoch., B. Normal Dist.	500	**1.53e+9**	**1.40e+9**	**1.30e+8**	**2.98e+6**	**552**	224.64
A STATUS QUO		750	1.25e+10	6.90e+9	5.59e+9	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	750	2.92e+9	2.40e+9	5.17e+8	5.10e+6	1468	16.00
LS	Class., $\hat{\beta}_{\text{avgday}}$	750	2.69e+9	2.18e+9	5.18e+8	4.63e+6	1470	51.88
LS	Stoch., B. Normal Dist.	750	**1.56e+9**	**1.39e+9**	**1.64e+8**	**2.97e+6**	**465**	299.99
A STATUS QUO		1000	1.44e+10	6.90e+9	7.45e+9	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1000	3.09e+9	2.41e+9	6.83e+8	5.12e+6	1453	16.04
LS	Class., $\hat{\beta}_{\text{avgday}}$	1000	3.00e+9	2.31e+9	6.83e+8	4.92e+6	1453	52.54
LS	Stoch., B. Normal Dist.	1000	**1.63e+9**	**1.39e+9**	**2.36e+8**	**2.96e+6**	**502**	117.19
A STATUS QUO		1500	1.81e+10	6.90e+9	1.12e+10	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1500	3.43e+9	2.42e+9	1.01e+9	5.14e+6	1438	15.80
LS	Class., $\hat{\beta}_{\text{avgday}}$	1500	3.36e+9	2.34e+9	1.01e+9	4.99e+6	1438	50.34
LS	Stoch., B. Normal Dist.	1500	**1.73e+9**	**1.39e+9**	**3.41e+8**	**2.96e+6**	**484**	231.40
A STATUS QUO		2179	2.31e+10	6.90e+9	1.62e+10	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	2179	3.89e+9	2.42e+9	1.46e+9	5.16e+6	1427	15.87
LS	Class., $\hat{\beta}_{\text{avgday}}$	2179	3.76e+9	2.30e+9	1.46e+9	4.88e+6	1427	47.10
LS	Stoch., B. Normal Dist.	2179	**1.91e+9**	**1.43e+9**	**4.85e+8**	**3.04e+6**	**474**	141.61
A STATUS QUO		4358	3.94e+10	6.90e+9	3.25e+10	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	4358	5.34e+9	2.45e+9	2.90e+9	5.21e+6	1414	15.91
LS	Class., $\hat{\beta}_{\text{avgday}}$	4358	5.21e+9	2.31e+9	2.90e+9	4.92e+6	1414	51.07
LS	Stoch., B. Normal Dist.	4358	**2.29e+9**	**1.45e+9**	**8.40e+8**	**3.09e+6**	**410**	314.67
A STATUS QUO		6538	5.56e+10	6.90e+9	4.87e+10	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	6538	6.79e+9	2.45e+9	4.34e+9	5.21e+6	1413	15.83
LS	Class., $\hat{\beta}_{\text{avgday}}$	6538	6.78e+9	2.43e+9	4.34e+9	5.18e+6	1413	51.20
LS	Stoch., B. Normal Dist.	6538	**2.71e+9**	**1.48e+9**	**1.23e+9**	**3.14e+6**	**401**	340.74
A STATUS QUO		33786	2.59e+11	6.90e+9	2.52e+11	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	33786	2.49e+10	2.46e+9	2.24e+10	5.23e+6	1411	15.73
LS	Class., $\hat{\beta}_{\text{avgday}}$	33786	2.48e+10	2.41e+9	2.24e+10	5.13e+6	1411	49.57
LS	Stoch., B. Normal Dist.	33786	**7.74e+9**	**1.66e+9**	**6.08e+9**	**3.54e+6**	**383**	127.63
A STATUS QUO		1.00e+6	7.46e+12	6.90e+9	7.45e+12	1.47e+7	15855	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1.00e+6	6.65e+11	2.46e+9	6.63e+11	5.23e+6	1411	15.88
LS	Class., $\hat{\beta}_{\text{avgday}}$	1.00e+6	6.65e+11	2.41e+9	6.63e+11	5.13e+6	1411	51.25
LS	Stoch., B. Normal Dist.	1.00e+6	**2.57e+11**	**1.74e+9**	**2.55e+11**	**3.71e+6**	**543**	89.99

optimization work in conjunction and allows for a realistic emission comparison of all solutions. The results are shown in Table 7.4.

An interesting aspect of the results is that the higher emission ratios do not always result in fewer disposed bins. This highlights that the employed methods are still heuristics and approximations. Interestingly, this limitation occurs in both the classical and stochastic

strategies, which implies that the data itself may also have less predictive qualities, i.e., observable patterns, than we initially expected.

Even so, all proposed methods dramatically outperform the status quo solution, especially when it comes to the number of disposed bins. This highlights the effectiveness of the solutions overall. Moreover, the stochastic local search consistently stands out, delivering the best results in both travel distance and the number of disposed bins.

Table 7.4: PCSP results: Summary of recycling efficiency and sustainability metrics based on actual pollution

Algorithm	Strategy	$\frac{\epsilon_{dispose}}{\epsilon_{collect}}$	Emissions [CO2e]			Distance	Disposal	Runtime
			f	Collection	Disposal	[m]	[stops]	[min]
A STATUS QUO		0	6.90e+9	6.90e+9	**0.00e+0**	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	0	1.46e+9	1.46e+9	**0.00e+0**	3.10e+6	**1848**	15.79
LS	Class., $\hat{\beta}_{\text{avgday}}$	0	**1.24e+9**	**1.24e+9**	**0.00e+0**	**2.64e+6**	**1848**	63.15
LS	Stoch., B. Normal Dist.	0	1.47e+9	1.47e+9	**0.00e+0**	3.14e+6	**1848**	151.92
A STATUS QUO		250	7.59e+9	6.90e+9	6.90e+8	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	250	2.54e+9	2.36e+9	1.80e+8	5.03e+6	1534	15.94
LS	Class., $\hat{\beta}_{\text{avgday}}$	250	2.29e+9	2.11e+9	1.80e+8	4.50e+6	1534	46.80
LS	Stoch., B. Normal Dist.	250	**1.54e+9**	**1.37e+9**	**1.65e+8**	**2.92e+6**	**1405**	218.82
A STATUS QUO		500	8.28e+9	6.90e+9	1.38e+9	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	500	2.79e+9	2.39e+9	4.07e+8	5.08e+6	1731	15.93
LS	Class., $\hat{\beta}_{\text{avgday}}$	500	2.54e+9	2.13e+9	4.07e+8	4.54e+6	1731	50.60
LS	Stoch., B. Normal Dist.	500	**1.73e+9**	**1.40e+9**	**3.28e+8**	**2.98e+6**	**1396**	224.64
A STATUS QUO		750	8.97e+9	6.90e+9	2.07e+9	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	750	3.01e+9	2.40e+9	6.10e+8	5.10e+6	1730	16.00
LS	Class., $\hat{\beta}_{\text{avgday}}$	750	2.78e+9	2.18e+9	6.10e+8	4.63e+6	1730	51.88
LS	Stoch., B. Normal Dist.	750	**1.90e+9**	**1.39e+9**	**5.04e+8**	**2.97e+6**	**1431**	299.99
A STATUS QUO		1000	9.66e+9	6.90e+9	2.76e+9	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1000	3.22e+9	2.41e+9	8.12e+8	5.12e+6	1729	16.04
LS	Class., $\hat{\beta}_{\text{avgday}}$	1000	3.13e+9	2.31e+9	8.12e+8	4.92e+6	1729	52.54
LS	Stoch., B. Normal Dist.	1000	**2.11e+9**	**1.39e+9**	**7.22e+8**	**2.96e+6**	**1536**	117.19
A STATUS QUO		1500	1.10e+10	6.90e+9	4.14e+9	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1500	3.64e+9	2.42e+9	1.22e+9	5.14e+6	1729	15.80
LS	Class., $\hat{\beta}_{\text{avgday}}$	1500	3.56e+9	2.34e+9	1.22e+9	4.99e+6	1729	50.34
LS	Stoch., B. Normal Dist.	1500	**2.48e+9**	**1.39e+9**	**1.08e+9**	**2.96e+6**	**1539**	231.40
A STATUS QUO		2179	1.29e+10	6.90e+9	6.01e+9	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	2179	4.19e+9	2.42e+9	1.77e+9	5.16e+6	1728	15.87
LS	Class., $\hat{\beta}_{\text{avgday}}$	2179	4.06e+9	2.30e+9	1.77e+9	4.88e+6	1728	47.10
LS	Stoch., B. Normal Dist.	2179	**2.89e+9**	**1.43e+9**	**1.47e+9**	**3.04e+6**	**1431**	141.61
A STATUS QUO		4358	1.89e+10	6.90e+9	1.20e+10	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	4358	5.98e+9	2.45e+9	3.54e+9	5.21e+6	1727	15.91
LS	Class., $\hat{\beta}_{\text{avgday}}$	4358	5.85e+9	2.31e+9	3.54e+9	4.92e+6	1727	51.07
LS	Stoch., B. Normal Dist.	4358	**4.25e+9**	**1.45e+9**	**2.80e+9**	**3.09e+6**	**1367**	314.67
A STATUS QUO		6538	2.49e+10	6.90e+9	1.80e+10	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	6538	7.75e+9	2.45e+9	5.31e+9	5.21e+6	1727	15.83
LS	Class., $\hat{\beta}_{\text{avgday}}$	6538	7.74e+9	2.43e+9	5.31e+9	5.18e+6	1727	51.20
LS	Stoch., B. Normal Dist.	6538	**6.39e+9**	**1.48e+9**	**4.91e+9**	**3.14e+6**	**1599**	340.74
A STATUS QUO		33786	1.00e+11	6.90e+9	9.32e+10	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	33786	2.99e+10	2.46e+9	2.74e+10	5.23e+6	1727	15.73
LS	Class., $\hat{\beta}_{\text{avgday}}$	33786	2.98e+10	2.41e+9	2.74e+10	5.13e+6	1727	49.57
LS	Stoch., B. Normal Dist.	33786	**2.66e+10**	**1.66e+9**	**2.49e+10**	**3.54e+6**	**1570**	127.63
A STATUS QUO		1.00e+6	2.77e+12	6.90e+9	2.76e+12	1.47e+7	5870	-
GREEDY	Class., $\hat{\beta}_{\text{avgday}}$	1.00e+6	8.14e+11	2.46e+9	8.12e+11	5.23e+6	1727	15.88
LS	Class., $\hat{\beta}_{\text{avgday}}$	1.00e+6	8.14e+11	2.41e+9	8.12e+11	5.13e+6	1727	51.25
LS	Stoch., B. Normal Dist.	1.00e+6	**7.20e+11**	**1.74e+9**	**7.18e+11**	**3.71e+6**	**1529**	89.99

CHAPTER 8

Conclusion

In this final chapter, we briefly summarize the thesis and then recapitulate how we answered the research questions. Finally, we give an outlook on potential future research directions and opportunities for extending the presented work.

8.1 Contributions of This Work

Contaminated waste is a problem in organic solid waste recycling, often leading to increased greenhouse gas emissions and disposal of resources that could otherwise have been recycled. To mitigate these issues, we proposed sorting the collection stops in advance based on historical pollution data, to separate polluted from clean waste. Our solution integrates predictive pollution modeling, tour optimization algorithms, and a knowledge graph-based framework tailored to the Pre-Collection Sorting Problem (PCSP). For the pollution prediction task, we applied statistical methods, knowledge graph embedding models, and stochastic models from both classical statistics and Bayesian statistics. We further proposed a greedy heuristic and an extensive local search algorithm for PCSP tour optimization.

Evaluated on six months of real-world data across 11 emission scenarios, our methods demonstrated a considerable reduction in disposed waste volume and emissions, achieving up to two-thirds improvement compared to status quo routes extracted from the data. These results show the practical impact of our approach as a contribution to the field of waste management and, more generally, efforts to transition to a circular economy.

Following, we recapitulate on the research questions and how they were addressed by this thesis.

Research Question 1. What is a suitable ontology for the domain of organic solid waste collection that supports pollution prediction and subsequent efficient tour

optimization?

In order to store data in different stages of the process, we proposed a layered ontology, containing a layer for raw data, one for preprocessed data, and multiple layers for algorithm-specific results. Each layer had different objectives, but as a general statement, we focused on a clear separation of concerns and query efficiency. For more insights into the individual layers, we refer the reader to Chapter 4.

Research Question 2. How can optimization algorithms be applied to waste collection tours to minimize operational costs and environmental impact while ensuring efficient separation of clean and contaminated waste?

We explored this question in three stages, first, we formalized the problem to establish a structure for further work in Chapter 3. There we also defined an objective function that contained components that are unknown at the time of route optimization. These components describe future pollution, which led us to propose two approximation strategies with numerous techniques in Chapter 5. Then, in Chapter 6, we explored algorithms for route optimization, backed up by SOTA vehicle routing problem literature.

Research Question 3. How can the effectiveness and efficiency of sustainable reasoning methodologies in waste collection optimization be evaluated, and what are appropriate metrics in this evaluation framework?

In Chapter 7, we designed and applied a framework for evaluating the proposed methods, including knowledge graph-based techniques. In short, we assessed the solution components separately and collectively, measuring prediction accuracy, route efficiency, and environmental impact.

8.2 Outlook

As this particular problem has, to the best of our knowledge, not been investigated in detail yet, there are numerous directions future work could follow. The following list provides some relevant options:

- We tested the proposed methods on one data set only and observed a high amount of noise. We attributed this at least partially to the required preprocessing, which could be reduced with either a different data collection method or artificial data.
- As shown in Figure 7.2, there is a considerable amount of pickup stops with only a few or no past reports that can be used for the predictions. A promising adaptation could involve sampling reports from the neighboring pickup stops.
- The knowledge graph embedding models may benefit from a differently structured ontology, that encodes relevant node and edge attributes such as the date of the report in the graph structure.

- Our selection of pollution prediction techniques is by no means extensive. Countless methods for this task remain, e.g., graph neural networks and various regression techniques.
- As vehicle routing is in general by no means static, e.g. due to traffic jams, another research direction could explore the adaptability of existing solutions to unexpected real-time changes.

Overview of Generative AI Tools Used

I used AI tools for feedback on this thesis, to improve the structure and formulations, and find potential issues with semantics and citations.

- Prompt: "In the following you find a master thesis. Please check the thesis for structural or semantic issues. Also regarding citations. Also, if you find really bad English formulations, please report it. Please provide detailed suggestions for each section."
- I additionally provided iterations of this master thesis as PDF documents.
- ChatGPT, model version GPT-4o, OpenAI, `https://chatgpt.com/`

List of Figures

List of Tables

List of Algorithms

Acronyms

AI artificial intelligence 9, 25

ATSP Asymmetric Traveling Salesman Problem 47

CE CROSS-exchange 47, 48

CO_2e CO_2-equivalent 25, 55, 56

GHG greenhouse gas 1–3, 5, 7, 16, 24, 25, 55

GNN graph neural network 12

GOO General Optimization Ontology 18

GUI graphical user interface 11

GVRP green vehicle routing problem 16

IoT Internet Of Things 3, 19

KG knowledge graph 3, 5, 6, 8, 9, 11–13, 15, 19, 33, 51, 54

KGE knowledge graph embedding 5, 9, 12–15, 29, 35–37, 54, 56, 57, 63, 69

LK Lin-Kernighan algorithm 18, 47

MDVRP multi-depot vehicle routing problem 16

ML machine learning 9, 13, 19, 20, 25

OSRM Open Source Routing Machine 27

OSW organic solid waste 2–4, 21, 24, 25, 43, 51, 55, 63

OWL Web Ontology Language 10, 11

PCSP Pre-Collection Sorting Problem 6, 7, 21, 23, 24, 27, 29, 30, 33–35, 43, 47, 49, 54, 59, 62, 63, 71

PDF probability density function 37, 38, 58

RDF Resource Description Framework 10

SOTA state of the art 5, 17, 37, 43, 64

TKGE temporal knowledge graph embedding 12

TSP Traveling Salesman Problem 18, 47

VRP vehicle routing problem 3, 8, 9, 16–19, 43, 44, 46, 49

Symbols

$b \in P$ base station 22, 23, 44–46

$\beta : R \cup (P \times D) \cup (\bigcup S_d) \mapsto \mathbb{R}$ pollution function 22, 24, 36, 38, 39, 53, 58, 69

β^{max} pollution threshold 24, 34, 35, 38

C vehicle capacity 22–24, 34, 44–46

D days 22, 23, 36

$\delta : L \times L \mapsto \mathbb{R}$ distance function 21–23, 44–46, 48

$e \in P$ waste drop-off station 22, 23, 44–46

$\epsilon_{collect}$ waste collection cost coefficient in gram CO_2e/m 24, 25, 34, 45, 46, 48, 54, 56

$\epsilon_{dispose}$ waste disposal cost coefficient in gram $CO_2e/pickup\ stop$ 25, 34, 45, 46, 48, 54, 56

$f : \bigcup S \mapsto \mathbb{R}$ objective function 59, 62

$\gamma : \Theta \mapsto T$ pollution type function 22

$\hat{\beta}$ approximation of β 35

$\hat{\beta}_{avg} : R \mapsto \mathbb{R}$ average pollution of past reports associated with report r 36, 57

$\hat{\beta}_{avgday} : R \mapsto \mathbb{R}$ average day pollution associated with report r 36, 57–59

$\hat{f}$ approximation of objective function 60

$\hat{\iota}_{stoch}$ approximation of ι in stochastic pollution model 38, 48

$\iota : \bigcup S_d \mapsto \{0, 1\}$ pollution type indicator function 25, 35, 38, 59

L locations 21

$\lambda : \bigcup S_d \mapsto \mathbb{R}$ route length function 23, 25

l_p location of pickup stop p 22

l_r location of report r 22

$\mathcal{C}_d$ pickup stops on day d classified as clean 35, 45, 46

$\mathcal{D}_d$ pickup stops on day d classified as polluted 35

P pickup stops 22, 23, 25, 26, 34, 38

P_d pickup stops occurring in day d 23, 35

$\pi : \Theta \mapsto [0, 1]$ tag confidence function 22

R waste collection reports 21, 22, 25

R_v^d route 23–25, 35, 38, 59

$\rho : R \mapsto P$ report clustering function 22–25, 34, 36

R^{PAST} past reports 22, 23

R_p^{PAST} past reports associated with pickup stop p 22, 23, 36, 38, 39

$S = (S_d)_{d \in D}$ PCSP solution 23, 25, 59

S_d routes of a day 23, 25

$\sigma : T \mapsto \mathbb{R}$ severity function 21, 22

T pollution types 21, 22

τ timestamps 21

τ' time of decision making 22

τ_r timestamp of report r 22, 36

$\Theta = \bigcup_{r \in R} \Theta_r$ pollution tags 22, 58

Θ_r pollution tags associated with report r 22

V waste collection vehicles 22, 23, 34

Bibliography

Anita Agardi, Laszlo Kovacs, and Tamas Banyai. Ontology support for vehicle routing problem. *Applied sciences*, 12(23):12299, 2022. ISSN 2076-3417.

Mohiuddin Ahmed, Raihan Seraj, and Syed Mohammed Shamsul Islam. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8): 1295, 2020.

Nadali Alavi, Khatereh Sarmadi, Gholamreza Goudarzi, Ali Akbar Babaei, Reza Bakhshoodeh, and Pooya Paydary. Attenuation of tetracyclines during chicken manure and bagasse co-composting: Degradation, kinetics, and artificial neural network modeling. *Journal of environmental management*, 231:1203–1210, 2019.

Juho Andelmin and Enrico Bartolini. An exact algorithm for the green vehicle routing problem. *Transportation Science*, 51(4):1288–1303, 2017.

Grigoris Antoniou and Frank van Harmelen. Web ontology language: Owl. *Handbook on ontologies*, pages 91–110, 2009.

Florian Arnold and Kenneth Sörensen. Knowledge-guided local search for the vehicle routing problem. *Computers & Operations Research*, 105:32–46, 2019.

Umweltbundesamt Austria. Emissionskennzahlen 2022. `https://www.umweltbundesamt.at/fileadmin/site/themen/mobilitaet/daten/ekz_fzkm_verkehrsmittel.pdf`, 2022. [Online; accessed 13.11.2024].

Nabila Azi, Michel Gendreau, and Jean-Yves Potvin. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, 41:167–173, 2014.

Franz Baader, Ian Horrocks, and Ulrike Sattler. Description logics. *Foundations of Artificial Intelligence*, 3:135–179, 2008.

Philippe Badeau, François Guertin, Michel Gendreau, Jean-Yves Potvin, and Eric Taillard. A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies*, 5(2):109–122, 1997.

Barrie M Baker and MA1951066 Ayechew. A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, 30(5):787–800, 2003.

Taimur Bakhshi and Muhammad Ahmed. Iot-enabled smart city waste management using machine learning analytics. In *2018 2nd International Conference on Energy Conservation and Efficiency (ICECE)*, pages 66–71, 2018. doi: 10.1109/ECE.2018.8554985.

Ivana Balažević, Carl Allen, and Timothy M Hospedales. Tucker: Tensor factorization for knowledge graph completion. *arXiv preprint arXiv:1901.09590*, 2019.

John E Bell and Patrick R McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced engineering informatics*, 18(1):41–48, 2004.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

Jose Caceres-Cruz, Pol Arias, Daniel Guimarans, Daniel Riera, and Angel A. Juan. Rich vehicle routing problem: Survey. *ACM Comput. Surv.*, 47(2), dec 2014. ISSN 0360-0300.

Linlin Chao, Jianshan He, Taifeng Wang, and Wei Chu. Pairre: Knowledge graph embeddings via paired relation vectors. *arXiv preprint arXiv:2011.03798*, 2020.

Geoff Clarke and John W Wright. Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568–581, 1964.

Marek Dudáš, Steffen Lohmann, Vojtěch Svátek, and Dmitry Pavlov. Ontology visualization methods and tools: a survey of the state of the art. *The Knowledge Engineering Review*, 33:e10, 2018. doi: 10.1017/S0269888918000073.

Lisa Ehrlinger and Wolfram Wöß. Towards a definition of knowledge graphs. *SEMANTiCS (Posters, Demos, SuCCESS)*, 48(1-4):2, 2016.

Eurostat. Treatment of waste by waste category, hazardousness and waste management operations, 2022a. URL `https://ec.europa.eu/eurostat/databrowser/product/page/ENV_WASTRT`.

Eurostat. Treatment of waste by waste category, hazardousness and waste management operations, 2022b. URL `https://ec.europa.eu/eurostat/databrowser/product/page/ENV_WASTRT`.

Dieter Fensel, Umutcan Simsek, Kevin Angele, Elwin Huaman, Elias Kärle, Oleksandra Panasiuk, Ioan Toma, Jürgen Umbrich, and Alexander Wahler. *Knowledge graphs*. Springer, 2020.

Martin Geissdoerfer, Paulo Savaget, Nancy MP Bocken, and Erik Jan Hultink. The circular economy–a new sustainability paradigm? *Journal of cleaner production*, 143: 757–768, 2017.

Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? *Handbook on ontologies*, pages 1–17, 2009.

Hao-nan Guo, Shu-biao Wu, Ying-jie Tian, Jun Zhang, and Hong-tao Liu. Application of machine learning methods for the prediction of organic solid waste treatment and recycling processes: A review. *Bioresource technology*, 319:124114, 2021.

M Arebey Hannan, Maher Arebey, Rawshan Ara Begum, and Hassan Basri. An automated solid waste bin level detection system using a gray level aura matrix. *Waste management*, 32(12):2229–2238, 2012.

Christiane Hannauer. Optimierung der sammlung und behandlung von grün-und bioabfällen aus der kommunalen sammlung in niederösterreich, 2014.

René Heinzl, Markus Nissl, and Emanuel Sallinger. Towards efficient annotation databases. In *Proceedings of the 15th Alberto Mendelzon International Workshop on Foundations of Data Management (AMW 2023)*., 2023.

Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000.

John Horváth. *Topological vector spaces and distributions*. Courier Corporation, 2012.

Valentina Janev, Damien Graux, Hajira Jabeen, and Emanuel Sallinger. *Knowledge graphs and big data processing*. Springer Nature, 2020.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and S Yu Philip. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514, 2021.

Zhuang Kang, Jie Yang, Guilan Li, and Zeyi Zhang. An automatic garbage classification system based on deep learning. *IEEE access*, 8:140019–140029, 2020.

Miyuru Kannangara, Rahul Dua, Leila Ahmadi, and Farid Bensebaa. Modeling and prediction of regional municipal solid waste generation and diversion in canada using machine learning approaches. *Waste management*, 74:3–15, 2018.

Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

Antoon WJ Kolen, AHG Rinnooy Kan, and Harry WJM Trienekens. Vehicle routing with time windows. *Operations Research*, 35(2):266–273, 1987.

Agustinus Kristiadi, Mohammad Asif Khan, Denis Lukovnikov, Jens Lehmann, and Asja Fischer. Incorporating literals into knowledge graph embeddings. In *The Semantic Web–ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part I 18*, pages 347–363. Springer, 2019.

Julien Leblay and Melisachew Wudage Chekol. Deriving validity time in knowledge graph. In *Companion proceedings of the the web conference 2018*, pages 1771–1776, 2018.

J. K. Lenstra and A. H. G. Rinnooy Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981. ISSN 0028-3045.

Bingjie Li, Guohua Wu, Yongming He, Mingfeng Fan, and Witold Pedrycz. An overview and experimental study of learning-based optimization algorithms for the vehicle routing problem. *IEEE/CAA Journal of Automatica Sinica*, 9(7):1115–1138, 2022.

Xiang Li, Qian Xie, Quanyin Zhu, Ke Ren, and Jizhou Sun. Knowledge graph-based recommendation method for cold chain logistics. *Expert Systems with Applications*, 227:120230, 2023. ISSN 0957-4174.

Shen Lin and Brian W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

John Miller, Gregory Baramidze, Amit Sheth, and Paul Fishwick. Investigating ontologies for simulation modeling. In *37th Annual Simulation Symposium, 2004. Proceedings*, pages 55–63, Los Alamitos CA, 2004. IEEE Computer Society. ISBN 9780769521107.

Aristide Mingozzi, Roberto Roberti, and Paolo Toth. An exact algorithm for the multitrip vehicle routing problem. *INFORMS Journal on Computing*, 25(2):193–207, 2013.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Sarah L Nordahl, Jay P Devkota, Jahon Amirebrahimi, Sarah Josephine Smith, Hanna M Breunig, Chelsea V Preble, Andrew J Satchwell, Ling Jin, Nancy J Brown, Thomas W Kirchstetter, et al. Life-cycle greenhouse gas emissions and human health trade-offs of organic waste management strategies. *Environmental science & technology*, 54(15): 9200–9209, 2020.

Sumit Pai and Luca Costabello. Learning embeddings from knowledge graphs with numeric edge attributes. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2869–2875. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/395. URL `https://doi.org/10.24963/ijcai.2021/395`. Main Track.

Jeff Z Pan. *Resource description framework*, pages 71–90. Springer, 2009.

Ted K Ralphs, Leonid Kopman, William R Pulleyblank, and Leslie E Trotter. On the capacitated vehicle routing problem. *Mathematical programming*, 94:343–359, 2003.

Yingtao Ren, Maged Dessouky, and Fernando Ordóñez. The multi-shift vehicle routing problem with overtime. *Computers & Operations Research*, 37(11):1987–1998, 2010.

David Rutqvist, Denis Kleyko, and Fredrik Blomstedt. An automated machine learning approach for smart waste management systems. *IEEE transactions on industrial informatics*, 16(1):384–392, 2019.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

Katherine Starr, Gara Villalba, and Xavier Gabarrell. Upgraded biogas from municipal solid waste for natural gas substitution and co2 reduction–a case study of austria, italy, and spain. *Waste Management*, 38:105–116, 2015.

Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: Principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

Mesut Toğaçar, Burhan Ergen, and Zafer Cömert. Waste classification using autoencoder network with integrated feature selection method in convolutional neural network models. *Measurement*, 153:107459, 2020.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction. In *International conference on machine learning*, pages 2071–2080. PMLR, 2016.

Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

Die Wiener Volkshochschulen. Abfallumrechnungstabelle. `https://www.umweltberatung.at/download/?id=abfallumrechnungstabelle-3044-umweltberatung.pdf`, 2023. [Online, accessed 13.11.2024].

Hanxi Wang, Jianling Xu, Haixia Yu, Xuejun Liu, Wei Yin, Yuanyuan Liu, Zhongwei Liu, and Tian Zhang. Study of the application and methods for the comprehensive treatment of municipal solid waste in northeastern china. *Renewable and Sustainable Energy Reviews*, 52:1881–1889, 2015.

WHO. Lead in drinking-water: health risks, monitoring and corrective actions: technical brief. In *Lead in drinking-water: health risks, monitoring and corrective actions: technical brief*. 2022.

Hailin Wu, Fengming Tao, and Bo Yang. Optimization of vehicle routing for waste collection and transportation. *International Journal of Environmental Research and Public Health*, 17(14):4963, 2020.

Zhi Xu, Bing Zhao, Yuyun Wang, Jinliang Xiao, and Xuan Wang. Composting process and odor emission varied in windrow and trough composting system under different air humidity conditions. *Bioresource Technology*, 297:122482, 2020.

Lantian Zhang, Guorui Liu, Sumei Li, Lili Yang, and Sha Chen. Model framework to quantify the effectiveness of garbage classification in reducing dioxin emissions. *Science of the Total Environment*, 814:151941, 2022.