MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

A Solution approach for the Periodic Vehicle Routing Problem in IoT-based Solid Waste Management

verfasst von / submitted by

Daniel Kapferer, B.Sc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2023 / Vienna 2023

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	UA 066 915
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Betriebswirtschaft
Betreut von / Supervisor:	Univ.-Prof. Mag. Dr. Karl Franz Dörner, Privatdoz.
Mitbetreut von / Co-Supervisor:	Alina-Gabriela Müller, BSc MSc PhD

Acknowledgement

I would like to express my gratitude towards Prof. Dörner for introducing me to Supply Chain Management and giving me the chance to write my master thesis at the Department of Business Decisions and Analytics. I further want to thank my supervisor Alina-Gabriela Müller for her continuous support and valuable suggestions. Moreover, I am grateful for the support of Alexander Kinast, who taught me coding in C# and who was a great mentor during my time as tutor in *Implementation of Optimization Techniques*.

I also appreciate the help from Dr. Fabien Tricoire very much, who helped me with setting up the instances for his tour visualization tool *proute*. It would have been a struggle without you.

Lastly, I would like to thank my girlfriend Sofia and my parents Susanna and Wolfgang Kapferer for their continuous support and belief in me, which allowed me to be motivated and focused on this endeavor. I am very grateful for you.

Vienna, March 2023

Daniel Kapferer

Abstract

This master's thesis addresses the Periodic Vehicle Routing Problem with deterministic and stochastic demands in the context of solid waste management, with a focus on incorporating sensor information from smart waste bins to improve routing schedules. Traditionally, solid waste collection methods rely on predetermined schedules without considering the filling levels of waste bins, resulting in inefficient routing schedules.

Therefore, the research questions in this study aim to determine, if sensors can be used to increase information and improve routing schedules, and if a limited number of sensors can be strategically placed to achieve these improvements.

The problem is implemented with C# and solved with a Tabu Search heuristic to generate feasible solutions. Instances established in Cordeau et al. (1997) are used to evaluate the results and are compared with generated instances using stochastic demands and varying degrees of sensor allocations.

The findings show the feasibility of using sensors connected to the Internet of Things (IoT) in solid waste management to improve routing schedules and provide valuable insights on optimal sensor placement strategies. The research can inform waste management practices and contribute to more sustainable and efficient waste collection systems.

Keywords: Periodic Vehicle Routing Problem, stochastic demand, Tabu Search, Solid Waste Management, IoT, Sensor Placement, Smart Bins

Zusammenfassung

Diese Masterarbeit befasst sich mit dem Periodic Vehicle Routing Problem mit deterministischen und stochastischen Bedarfen im Kontext der Abfallwirtschaft. Dabei liegt der Schwerpunkt auf der Inklusion von Sensorinformationen aus intelligenten Abfallbehältern zur Verbesserung der Tourenplanung. Traditionell beruht die Routenplanung in der Abfallsammlung auf vorgegebenen Plänen, ohne den Füllstand der Abfallbehälter zu berücksichtigen, was zu ineffizienten Tourenplänen führen kann.

Die Forschungsfragen dieser Studie zielen daher darauf ab, festzustellen, ob Sensoren eingesetzt werden können, um den Informationsgehalt zu erhöhen und die Tourenpläne zu verbessern. Außerdem soll erforscht werden, ob eine begrenzte Anzahl von Sensoren strategisch platziert werden kann, um Tourenverbesserungen zu erreichen.

Das Problem wird mit C# implementiert und mit einer Tabu-Suche gelöst. Dabei werden Instanzen aus Cordeau et al. (1997) zur Bewertung der Ergebnisse herangezogen und mit generierten Instanzen verglichen, die stochastische Bedarfe und unterschiedliche Mengen an Sensoren verwenden.

Die Ergebnisse zeigen die Möglichkeit des Einsatzes von Internet of Things (IoT)-Sensoren in der Abfallwirtschaft zur Verbesserung der Tourenplanung und liefern wertvolle Erkenntnisse über optimale Strategien zur Platzierung der Sensoren. Die Forschungsergebnisse können Einfluss auf die Praxis in der Abfallwirtschaft haben und zu nachhaltigeren und effizienteren Abfallsammelsystemen beitragen.

Schlüsselwörter: Periodic Vehicle Routing Problem, stochastischer Bedarf, Tabu-Suche, Abfallwirtschaft, IoT, Sensorplatzierung, intelligente Mülltonnen

1 Table of Contents

List of Abbreviations

ALNS	Adaptive Large Neighborhood Search
ASC	Ascending
CVRP	Capacitated Vehicle Routing Problem
DESC	Descending
DTD	Detour-To-Depot
IoT	Internet of Things
PVRP	Periodic Vehicle Routing Problem
PVRPSD	Periodic Vehicle Routing with Stochastic Demands
REC	Reoptimization after Every Customer
RDF	Reoptimization after Delivery Failure
SPR	Stochastic Program with Recourse
SPRP	Periodic Routing Problem with Stochastic Demand
SVRP	Stochastic Vehicle Routing Problem
SWM	Solid Waste Management
VRP	Vehicle Routing Problem
VRPSD	Vehicle Routing Problem with Stochastic Demands
WBARP	Waste Bin Allocation and Routing Problem

List of Figures

List of Tables

1 Introduction

In this chapter the problems and inefficiencies of current solid waste management (SWM) associated with the collection of waste are established. Subsequently, the sensor-based solid waste management is introduced, that lays the foundation for a more efficient solid waste collection and serves as reference point for the Periodic Vehicle Routing Problem (PVRP) with deterministic and stochastic demands.

Current waste management systems are often inefficient because the filling level of containers is not known before departure and therefore either unnecessary collections are done, or the vehicle capacity is not sufficient to gather the remaining waste. If sensors were to measure the amount of waste inside the bins, this information could be used to design an efficient collection route using algorithms used in operation research (Morais et al. 2022). Based on the idea of adding sensor to waste bins to increase to operational efficiency, SWM based on the Internet of Things (IoT) can be a possible solution. IoT has an enormous potential to benefit SWM by providing real-time data through sensors equipped in the waste bins, which in turn can lead to better suited routes for waste collection. Usually, waste bin levels are not known a-priori and thus are stochastic. Through the introduction of sensors, waste bin levels are known before visiting, thus unnecessary visits because of low waste levels and cases in which the capacity of the vehicle would be exceeded, can be prevented. Consequently, this results in decreases in routing costs and a lower environmental impact associated with less fuel consumption.

In the literature there are a large number of articles where IoT-based SWM is being referred to and its potential improvements have been shown or praised. One way to model the collection of waste in SWM is through the use of the Periodic Vehicle Routing Problem, as in waste collection problems, customers need to be served periodically in a given period, for example once or twice a week and not on the weekend. One caveat of applying the PVRP to trash collection in SWM is, that the waste treatment plant cannot usually be modelled and the vehicles are usually returning to the depot when their shift is over (Angelelli & Grazia Speranza, 2002). Nevertheless, the PVRP is well suited for modelling the collection of waste and has been used extensively in the literature for this purpose.

For this work, the PVRP will be the problem formulation of choice and a heuristic algorithm will be used to solve PVRP instances from the literature and to test whether strategic placement of IoT-sensors on the waste bins will lead to a reduction in tour lengths and more efficient operation.

In the following Chapter 2, the underlying basic problem in the form of the Vehicle Routing Problem (VRP) and the closely related problems Capacitated Vehicle Routing Problem (CVRP) are explained at first. Going forward, the focus moves to the PVRP and the Periodic Vehicle Routing Problem with stochastic demands (PVRPSD). The Model formulations are shown in Chapter 3, followed by the main part consisting of the solution framework, in which the Tabu Search Algorithm, used for solving PVRP instances with deterministic demands, is explained in detail. Moreover, the adaptions and reoptimization procedures for solving the PVRPSD coupled with a strategic sensor placement will be discussed in Chapter 4. Finally, in Chapter 5, the solutions for the PVRP and PVRPSD with varying number of sensors are presented and the results for different approaches to allocating sensors to waste bins strategically are laid out.

2 Literature Review

To understand the complex problem of smart solid waste collection with sensor-support, a few essential topics need to be addressed first. Starting with Chapter 2.1 the fundamentals of vehicle routing are explained and the current models regarding this topic are discussed. In Chapter 2.2 solution methods for the underlying problem are covered with an emphasize on heuristic solution algorithms, specifically focusing on the Tabu Search metaheuristic, which will be used as a mean to solve instances of the Periodic Vehicle Routing Problem in the implementation chapter. Finally, in Chapter 2.3 concepts regarding the smart waste bin allocation are examined. The chapter starts with a review on how stochasticity can be assigned in the models. Afterwards an overview of the use of smart waste bins and the Sensor Placement Problem is given, in which numerous ways are discussed how the strategic placement of sensors can benefit vehicle routing based on current literature.

2.1 Fundamentals of Vehicle Routing

When it comes to problems, which involve the delivery of goods from storage facilities to customers, usually these are defined as Vehicle Routing Problems (Tan & Yeh, 2021). Since the first proposal of the VRP by Dantzig and Ramser (1959), who modelled an optimal routing of gasoline delivery trucks with one central depot and several gas stations to deliver to, a lot of research has been conducted in the field of vehicle routing. Since then, a plethora of variants of VRPs have been developed.

This chapter focuses on the CVRP, as well as the PVRP with deterministic and stochastic demands specifically. While Chapter 2.1.1 will provide the foundation for explaining the VRP and its basic restrictions based on the CVRP, Chapter 2.1.2 lays down the fundamentals for the PVRP, where several periods are considered, and customer service demand quantities are known beforehand. Chapter 2.1.3 discusses the Stochastic Vehicle Routing Problem (SVRP) and expands the basic model to include unknown customer demand quantities. Finally in Chapter 2.1.4 the combination of the PVRP and SVRP is discussed.

2.1.1 The Capacitated Vehicle Routing Problem

The Vehicle Routing Problem is defined as a problem, in which routes starting from one or several depots have to be designed in a way as to deliver or collect goods to/from geographically dispersed customers or cities optimally, while considering side restrictions (Laporte, 1992).

One of the most common forms of Vehicle Routing Problem, that is often encountered by firms operating in the distribution or logistic sector is the Capacitated Vehicle Routing Problem, which is not only the first problem explored in the literature and applied in practice, but also the one studied the most (Kim et al. (2015); Mańdziuk and Nejman (2015), as cited in (Konstantakopoulos et al., 2022). According to Toth and Vigo (2014) the CVRP is characterized by its homogenous vehicle fleet with identical capacity restrictions, a single depot and deterministic demands that cannot be split and are known before execution. Usually, the objective consists of minimizing the routing costs. Further it is usually assumed, that the travel distances are symmetrical between the customers, but asymmetrical formulations are also present. There are many adaptions possible to this basic Vehicle Routing Problem, however covering all forms is outside the bound of this thesis, therefore only the problems most suitable VRPs for this work are introduced.

2.1.2 The Periodic Vehicle Routing Problem

The Periodic Vehicle Routing Problem is a special case of the Vehicle Routing Problem, which involves visiting a predetermined number of customers by a vehicle one or more times in a given time-frame (Golden et al., 2008). Each customer has a set of possible visiting schedules, which is not set beforehand. For each problem, a number of vehicles are available to serve the customers and each vehicle needs to leave the depot, visit a number of customers, and returns to the depot, when its capacity is exhausted. The main objective of this problem is to minimize the travelling distance of the vehicles in the given time frame, while the capacity, vehicle amount and schedule restrictions need to be fulfilled (Angelelli & Grazia Speranza, 2002). To solve this problem adequately, visiting schedules need to be assigned for each customer and subsequently tours need to be created for the customers assigned to the given days so that all customers are served. Essentially a CVRP needs to be solved for each day.

The Periodic Vehicle Routing Problem is formulated by Cordeau et al. (1997) on a multigraph G = (V, A), consisting of a set of nodes $V \{v_o, v_1, \ldots, v_n\}$ and a set of edges $A \{(v_i, v_j)^{k,l}\}$, where k refers to the number of vehicles and l refers to the number of days. Additionally, the nonnegative costs c_{ijkl} corresponding to each arc $(v_i, v_j)^{k,l}$ are of importance, as they represent the travel times from customer i to customer j with vehicle k on day l. To formulate the PVRP as binary linear problem, several additional constants and variables have to be introduced. The binary variables x_{ijkl} is set to 1, only when customer j is visited by vehicle k on day l directly after visiting customer i, with the prerequisite, that customer i cannot be customer j.

Furthermore, the constant a_{rl} is equal to 1, whenever the day l is part of the visit combination r. Moreover, the binary constant y_{ir} is set to 1 in the case that the customer i is assigned the schedule $r \in C_i$. Finally, d_0 and q_0 must be equal to 0 and the PVRP can be formulated in the following:

Objective Function:

$$Min \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m}\sum_{l=1}^{t} c_{ijkl}x_{ijkl}, \tag{1}$$

subject to:

$$\sum_{r \epsilon c_i} y_{ir} = 1 \; (i = 1, \ldots, n); \tag{2}$$

$$\sum_{j=0}^{n} \sum_{k=1}^{m} x_{ijkl} - \sum_{r \epsilon c_i} a_{rl} \, y_{ir} = 0 \tag{3}$$

(i = 1, . . . ,n; l = 1, . . . , t)

$$\sum_{i=0}^{n} x_{ihkl} - \sum_{j=0}^{n} x_{hjkl} = 0 \tag{4}$$

(h = 0; . . . , n; k = 1, . . . , m; l = 1, . . . , t);

$$\sum_{j=1}^{n} x_{0jkl} \leq 1 \; (k = 1, \ldots, m; l = 1, \ldots, t); \tag{5}$$

$$\sum_{i=0}^{n} \sum_{j=0}^{n} q_i x_{ijkl} \leq Q_k \tag{6}$$

(k = 1, . . ., m; l = 1, . . . , t);

$$\sum_{i=0}^{n} \sum_{j=0}^{n} (c_{ijkl} + d_{i}) \, x_{ijkl} \leq D_k \tag{7}$$

(k = 1, . . . m; l = 1, . . . , t);

$$\sum_{v_i \in S} \sum_{v_j \in S} x_{ijkl} \leq |S| - 1 \tag{8}$$

(k = 1, . . . , m; l = 1, ... t;

S ⊆ V \ {0}; |S| ≥ 2);

$x_{ijkl} \in \{0,1\} \; (i = 0, . . . , n; j = 0, \ldots, n;$

(k = 1, . . . m; l = 1, . . ., t); (9)

$\boldsymbol{y_{ir}} \in \{0,1\}$ ($i = 1$, . . . , n; r ∈ C_i) (10)

Note: the formulation of the PVRP has been reprinted from Cordeau et al. (1997)

The objective function minimizes the total transportation cost incurred by visiting all customers, while the constraints guarantee feasibility. Constraint (2) ensures, that only feasible visit combinations can be chosen for visiting customers, while constraint (3) enforces only feasible days of visits for the given schedule. Whenever a vehicle arrives at a customer, a departure on the same day needs to be ensured, which is imposed by (4). Furthermore, vehicles can only be used once for each day (5). Additionally, the capacity restriction of the vehicles as well as the duration restriction for each route are enforced by constraint (6) and (7). Lastly, common subtour eliminations are addressed by (8). In (9) and (10), the decision variables for customer visits and customer schedules are depicted.

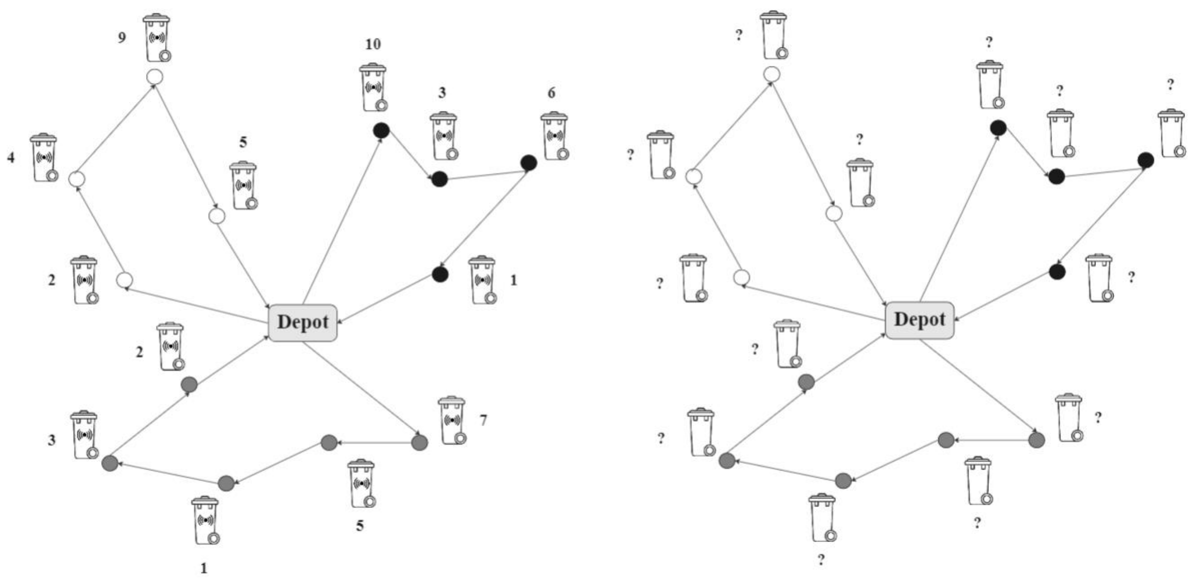
2.1.3 The Stochastic Vehicle Routing Problem

A Stochastic Vehicle Routing Problem occurs in cases, where certain aspects of the underlying VRP are created at random. Usually, random components in this problem could be the travel time, the existence of customers or their demand (Gendreau et al., 1996). Oyola et al. (2018) refer to these components as source of stochasticity and are considered to be an important feature connecting most articles on SVRPs. In this thesis the sole source of stochasticity is based on the demand. In this regard, Vehicle Routing Problems with stochastic demand (VRPSD) are closer to most real-world applications, where customers have stochastic demands. This problem can arise in several situations including the collection of solid waste, where the waste levels at each bin are unknown (Yang et al., 2000). Yang et al. (2000) explain, that in the VRPSD tours cannot always be conducted as previously planned due to the uncertainty of the demand, which can lead to situations where demands are not satisfied and actions need to be enforced to guarantee feasibility of the VRPSD.

In Figure 1, a VRPSD in solid waste management is visualized. On the left side, all bins are equipped with IoT sensors connected to the allowing for real-time delivery of (deterministic) filling levels. On the right side, the same routing problem is represented, the only difference is the lack of sensors equipped to the bins. As a result, the filling levels are not known in real-time. For this reason, the filling levels need to be estimated based on historical data, which are modelled as stochastic because filling amounts usually vary between certain levels.

Figure 1

Vehicle Routing Problem in solid waste management



Note: bins with Sensors (left); bins without Sensors (right); own illustration

To solve VRPs with stochastic demand, two main modelling choices for the VRPSD can be distinguished, which are chance-constrained approaches and recourse approaches (Ak & Erera, 2007). In the chance-constrained-programming (CCP) method for the VRPSD, the goal is to create routes in such a way, that failure probabilities are kept low (Florio et al., 2020). Usually there is a deterministic objective and route failure costs are often ignored (Oyola et al., 2018). The second modelling choice is the stochastic program with recourse (SPR), in which route failures are permitted, however a recourse policy needs to be executed, that specifies how the solution has to be repaired after a delivery failure. The objective consists of minimizing the sum of the traveling costs including the costs of any retours (Oyola et al., 2018).

According to Gendreau et al. (1996) (as cited in Oyola et al. (2018)), SPRs are harder to solve, but have a more relevant objective. This thesis uses the SPR modelling paradigm since routing decisions do not need to be specified before execution and penalties for overcapacity or overduration are allowed.

In the SPR usually two approaches are used for solving the VRPSD namely the a-priori optimization and the reoptimization. In the former case, a two-stage model is utilized. In the first stage preliminary tours are constructed, followed by the second stage, in which the demands are revealed, and recourse actions need to be conducted based on a previously determined policy (Toth & Vigo, 2014). In the latter case, tours can be reoptimized whenever the demand is revealed, by reordering unvisited customers in order to improve the given solution (Laporte et al., 2002). When comparing these two methods, a distinct difference can be identified:

> While a priori optimization is dominated in terms of solution quality by a reoptimization strategy, it is a preferable approach from a computational point of view since it entails solving only one instance of an NP-hard problem. (Laporte et al., 2002, p. 416).

It comes as no surprise, that the reoptimization in which customers are rearranged, takes much longer, but usually produces better results. Still, the a-priori approach is ideally suited as a basis for comparison and delivers fast results.

The most simplistic recourse policy detour-to-depot (DTD) uses routes, which were designed a-priori in respect to the ordering of potential customers to visit. Then the vehicles are driving to the customers in the previously determined order and whenever the capacity is reached, the vehicle has to drive back to the depot and unload. As noted in Florio et al. (2020) this recourse policy is not ideal, as restocking is only done, when the vehicle's capacity is exceeded. This applies particularly for customers far from the depot.

Another possibility is to introduce an optimal restocking policy, which was initially established by Yee and Golden (1980). In this improved approach pre-emptive detours to the depot are done before overcapacity can occur, so that potential tour failures at a later stage are prevented (Florio et al., 2020).

In contrast, routes in reoptimization models are not fully determined during the tour planning, but rather evolve when added information is obtained. This usually happens when a new customer has been visited and their demand becomes known (Florio et al., 2020). According to Secomandi and Margot (2009) the reoptimization approach in vehicle routing involves "a routing and a replenishment decision in a dynamic fashion: given the available vehicle capacity and the set of unserved customers, s/he decides which customer to visit next, whether directly or by performing an en route replenishment at the depot" (Secomandi & Margot, 2009, p. 214).

To see if the more complex reoptimization procedure produces better results, detour-to-depot and two forms of reoptimization will be compared in the implementation of the Tabu Search algorithm in Chapter 4 to solve the Periodic Vehicle Routings Problems with stochastic demands, where detours will be necessary. In the following this problem will be discussed.

2.1.4 The Periodic Vehicle Routing Problem with Stochastic Demand

The Periodic Vehicle Routing Problem with stochastic Demand (PVRPSD) is where the VRPSD and the PVRP overlap. It incorporates the multiperiod planning horizon of the PVRP and the stochastic demand, which can be attributed to the VRPSD. In previous research the abbreviation PVRPSD has not yet been encountered and Campbell and Wilson (2014) mention, that only a limited number of articles deal with this topic. One example that is named in the article by Campbell and Wilson (2014) is the Periodic Routing Problem with Stochastic Demand (SPRP) discussed by Schedl and Strauss (2011), which is also dealing with a multiperiod vehicle routing problem and customers with stochastic demands. Due to the combination of stochastic and multiperiod elements the model is more refined and resembles certain aspects of the real word better. For example Dayarian et al. (2016) are using it to model a multi-period milk collection problem, in which the stochastic demand underlies a seasonal component. However, the stochastic demand component does not have to be coupled to a periodic influence but is also suitable to represent uncertainty in a shorter time span, as it can be seen in waste collection, in which filling levels of trash bins are fluctuating.

In the PVRPSD solution of this thesis the values of the PVRP are used as an a-priori solution and then the solution is adapted, when the demand is revealed. For this reason, it can be considered a two-stage solution process.

In conclusion a plethora of possible adaptions to the classical VRP are used in current research, which can be tailored to the underlying problem. Due to the characteristics of solid waste collection discussed in this thesis, combining aspects from the PVRP and the SVRP is most suitable and will be adopted in the further course of this work.

2.2 Heuristic Solution Methods

In this chapter heuristic solution methods are introduced, which are used to solve several forms of VRPs. At first, the difference between exact and heuristic solution methods is discussed and reasons for and against the use of heuristics methods are examined. Then, the constructive heuristic is presented in Chapter 2.2.1, that enables the generation of a starting solution in the VRP. Finally, the Tabu Search Metaheuristic is reviewed in Chapter 2.2.2, which will be an integral part of solving the Periodic Vehicle Routing Problem.

2.2.1 Heuristic vs Exact Solution Methods

A recent survey by Tan and Yeh (2021) examined vehicle routing literature between 2019 and 2021 with a taxonomic framework and classified the solution algorithms into the categories heuristic- and metaheuristic algorithms as well as exact algorithms.

Exact solution methods can solve problems to optimality, however these sophisticated mathematical algorithms can only solve small problem sets consisting of around 100 customers (Laporte et al., 2014). Yet, Laporte et al. point out, that real-world scenarios are usually much larger and need to be solved in a short time span, which is not feasible with exact algorithms. Additionally, exact algorithms have the disadvantage of being quite inflexible, as they differ for each problem setting. Therefore, heuristics are needed to tackle the issues of exact solution methods and allow for solving large instances to a satisfactory level in a comparatively short time coupled with a certain degree of flexibility.

Tan and Yeh (2021) conclude, that heuristic and metaheuristic algorithms are still the main solution methods used in solving various types of vehicle routing problems. However, the authors also state that exact methods are catching up and speculate that this might be due to the rapid increase in computational power and memory power that allows for solving larger instances in faster times.

According to Silver (2004, p. 939), there are seven general forms of heuristic methods that are covered by “Randomly generated solutions, Problem decomposition/partitioning, Inductive methods, Methods that reduce the solution space, Approximation methods, Constructive methods and Local improvement (neighborhood search) methods”. The author notes that the methods are often combined and are not necessarily distinct from one another.

In the following, the construction heuristic for the initial solution generation is presented, followed by the metaheuristic Tabu Search, which combines several aspects of the mentioned categories of heuristics and uses the initial solution produced by the construction heuristic as starting solution.

2.2.2 Tour Construction Heuristics

According to Silver (2004) construction heuristics utilize the data given for the problem set in order to build an initial solution. For vehicle routing problems with a single vehicle as in the Travelling Salesman Problems (TSP), the Greedy Algorithm (GR) and Nearest Neighbor Algorithm (NNA) are commonly used greedy constructive heuristics, as they have shown to deliver good results for TSPs with euclidean distances, but do not perform well ins TSPs with symmetry or asymmetry (Gutin et al., 2001). In a greedy algorithm, “One would choose in each step the solution component with the greatest impact on the target parameter” (Zäpfel et al., 2010, p. 10). However, this approach has some drawbacks, as greedy heuristics usually produce only a small number of distinct solutions. Furthermore, the final solution quality is often inhibited by the shortsighted decisions made in the initial phase, causing bad moves in the final solution stage (Gendreau & Potvin, 2010, p. 229).

Consequently, for the purpose of achieving good computational results, solely relying on constructive heuristics is not enough. Instead, they serve as a starting point for generating better results via local searches and improvement heuristics. More refined solutions can be achieved by metaheuristics, which are advanced heuristics, allowing the combination of local searches and strategies to overcoming local minima (Gendreau & Potvin, 2010, p. ix).

One of these metaheuristics, which can be used to solve PVRPs will be introduced in the following chapter.

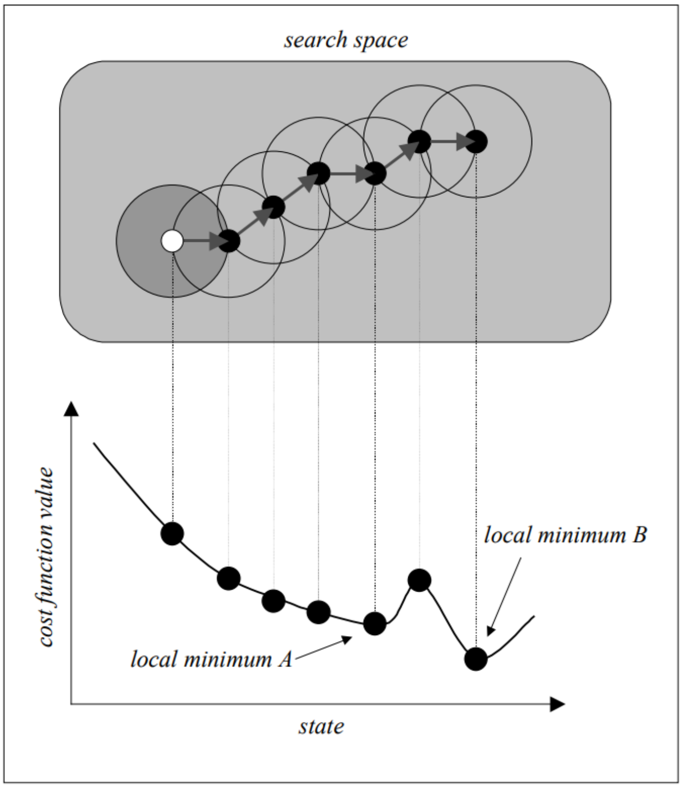
2.2.3 Tabu Search Metaheuristic

Tabu Search is a metaheuristic, which is used as a framework for a wide variety of strategies using iterative local search in the realm of discrete optimization (Gendreau & Potvin, 2010). It is built on the notion, that for solving a problem intelligently, the use of an adaptive memory and responsive exploration is needed in order to traverse the solution space efficiently and effectively (Glover & Laguna, 1998).

Its fundamental principles have first been formalized by Glover (1989) and consists of mainly short-term memory processes coupled with intermediate and long-term memory processes that allow for intensification and diversification of the search (Glover, 1989, p. 190). The two key components of Tabu Search are the forbidding of moves, also called Tabus and the freeing of the search. Tabus are used to restrict the search space, while the freeing of the search allows forgetting moves in a strategic fashion, through the use of short-term memory functions (Glover, 1989). This technique allows to escape local minima and stands in contrast to common greedy heuristics like hill-climbing, in which the solution gradually finds better solutions at each iteration but comes to a halt as soon as a local optimum is encountered. Figure 2 shows that Tabu Search can find a local minimum and overcome it by using non-improving moves coupled with tabu moves for cycle prevention, which than can lead to a new search space and allows for finding new local minima.

Figure 2

Neighborhood search in Tabu Search algorithms



Note: Figure 2 from Borghetti et al. (2001)

To prevent cases, in which the current best solution is worse than the previously best solution, a guidance procedure is necessary, but this comes with the risk of creating a cycle of reoccurring solutions. While metaheuristics like Simulated Annealing allow moves that deteriorate the current solution with a certain probability, Tabu Search makes use of tabus and penalizing moves in order to prevent cycles that would lead to solutions that have been explored a few iterations ago (Glover & Taillard, 1993). Due to the restriction of recent moves, the neighborhood is constantly changing, thus Tabu Search is considered to be a dynamic neighborhood method (Glover & Laguna, 1998).

The neighborhood restrictions are enforced by tabus, which are used to prevent cycling of moves. This is achieved by forbidding recent moves, which are put in a tabu list, that is permanently updated (Gendreau et al., 1994). If the solutions are in the tabu list, the solutions will not be considered as potential new solutions for the current iterations, except in cases where an aspiration criterion is set. Additionally, if the solution in the tabu list has a value below the aspiration level, which often means that the value is deemed as the current best solution, then the solution will be considered in this event (Bräysy & Gendreau, 2002).

Tabu Search can be adapted in many ways, but one element that still needs to be discussed, as it will be implemented in the solution framework for solving the PVRP, is the use of diversification. Instead of focusing more closely on solutions that have been deemed good in the past and seeking for nearby solutions, the search space is manipulated in a way that less frequently used attributes are incorporated, thus resulting in a more heterogenous search space with potentially good new solutions (Glover & Laguna, 1998).

One of these diversification measurements implemented, is the use of a frequency-based approach, which complements the recency-based tabu list quite well. While recency-based approaches concentrate on the recent moves, which leads them to having a short-term focus, frequency-based approaches are most effectively used in longer term strategies and can be implemented to ensure continuous diversification. As Soriano and Gendreau (1996) explain, in this diversification method historical information is used to steer the search continuously to new solutions by penalizing solutions, that have been visited often. This means that solutions, which consist of elements, which were already often contained in the best solution, receive a higher penalty, so that new solution spaces can be explored.

After addressing the heuristic solution methods in general and discussing Tabu search in the context of solving the PVRP with deterministic demands, the following chapter will focus on the allocation of smart waste bins, which is needed to solve the PVRPSD.

2.3 Smart Waste Bin Allocation

To solve the PVRPSD, as it is encountered in solid waste management, certain aspects from the literature have to be clarified. One special consideration that differentiates the PVRP with deterministic demands, and the PVRP with stochastic demands is the assignment of stochasticity. As there are diverse ways to do the assignment, Chapter 2.3.1 will discuss the individual methods. Then, in Chapter 2.3.2 the use of smart waste bins in solid waste management and their potential benefit is reviewed. Finally, in Chapter 2.3.2 the sensor placement problem is examined and examples regarding the ways of allocation of a limited number of sensors and their expected benefits are given.

2.3.1 Distribution Functions for Assigning Stochastic Demand

The PVRP instances in the body of research are generally designed with deterministic demands and are suitable to model the routes in a multi-period solid waste collection problem with a given customer schedule, where full information about the customers/waste bins is given. However, the deterministic demand does not appropriately capture the varying filling levels of waste bins in SWM, which would be better modelled by a stochastic demand.

For this reason, several distribution functions are used to model the varying demands in the VRPSD. Depending on whether the customer demand is based on a continuous or discrete probability function, the distribution functions and results differ. Due to the structure of the customer demands in the original data set, in which customers' demands are modelled as integer values, the most suitable distribution function is the one that can easily replicate this format for the VRPSD. Because of this, discrete distribution functions are the perfect fit. According to Oyola et al. (2018) various discrete probability functions have been used in the literature for the Capacitated Vehicle Routing Problem with Stochastic Demands (CVRPSD), namely discrete uniform, binomial, discrete triangular or Poisson Distribution functions. In the body of research, the Poisson Distribution function has been used in many articles for the purpose of assigning stochastic demand (Christiansen et al., 2009; Florio et al., 2022; Gauvin et al., 2014; Goodson et al., 2012; Laporte et al., 2002). As this has method has proven to produce suitable demand distributions, the Poisson Distribution function will be used in this thesis for the stochastic demand assignment. The exact approach of assigning stochastic demand with the Poisson distribution will be explained in Chapter 3.4.1.

While these probability distributions used in the literature are easy to handle and trace, it has been noted by Gendreau et al. (2016) that some assumption about the probability distribution are simplified too much. For one, the stochastic parameters in the literature are often uncorrelated, while this is usually not the case in real-world situations. Secondly, the range of probability distributions used in the literature has not yet been exhausted, as many distribution functions comply with the convolution principle, thus limiting their possible varieties. Yet, due to the convenience of implementing uncorrelated demands and the scope of this work, it is the method of choice for this thesis.

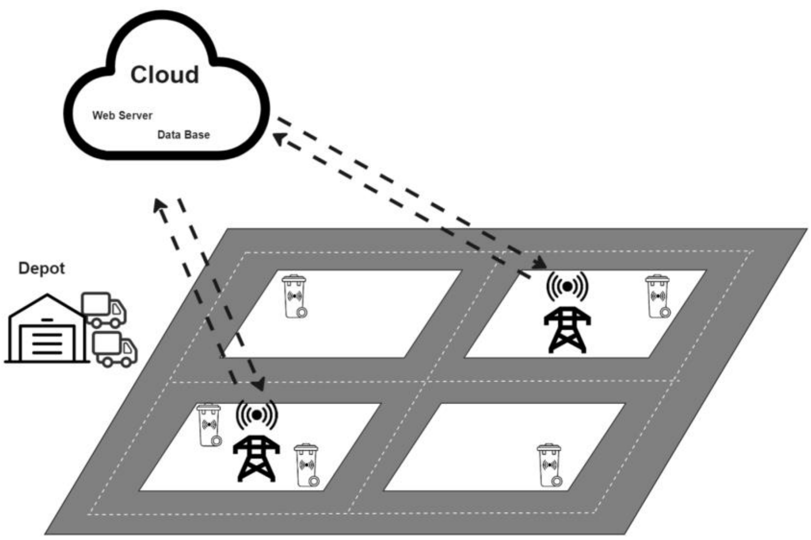
2.3.2 The Use of Smart Waste Bins in Solid Waste Management

In the literature a large number of articles are investigating the use of smart waste bins. In the survey by Anagnostopoulos et al. (2017) challenges and opportunities in IoT-based solid waste management are addressed. The authors highlight the potential of adding sensors to bins in SWM and refer to their use as key enabler in current waste management. Similarly, Esmaeilian et al. (2018) focus on possible ways of improving SWM and making it future-proof. One key element the authors highlight in their framework of waste collection is the use of sensors in the waste infrastructure combined with timely collection and recovery of waste. However the authors criticize, that waste management is often not seen holistically, but rather focuses on the traditional aspect of collecting waste without considering the whole product lifecycle.

In a study by Mishra and Ray (2020) a cyber-physical system for route optimization in SWM in Bhubaneswar is proposed. In this study waste bins equipped with sensors combined with real-time traffic information were used to reduce the capital expenses and operational expenses. The simulation allowed to decrease not only the total tour length but also the amount of vehicles used and decreased the overall cost by about 30%. An overview of such a system can be seen in Figure 3, that depicts roads in a city with smart bins and the necessary cloud infrastructure.

Figure 3

Overview of an IoT-based solid waste management Infrastructure



Note: This depiction is based on Figure 1 in Omara et al. (2018)

In a similar setting, Johansson (2006) investigate the use of dynamic routing and scheduling in SWM to determine, whether costs, distances and labor hours could be reduced in comparison to a predetermined schedule. Based on an empirical simulation study using 3.000 sensor-equipped recycling bins in a municipality in Sweden, simulations and analytical models have been performed. The authors conclude that the operating cost, number of containers collected, and distance driven, could be reduced. In settings where the demand is irregular, cost could be decreased the most. However, for a lower number of sensors or a shorter distance between the containers, the benefits decrease sharply. Additionally, the investment cost of the sensors has not been factored in and the authors mention that an ideal way of using sensor data for the planning phase has not yet been identified.

In a subsequent study, Johansson and Johansson (2008) also found out, that the policy *dynamic scheduling and dynamic routing to full containers* showed superior performance in larger systems, while the policy *dynamic scheduling and dynamic routing to "almost" full containers* fared better in smaller systems, resulting in cost reductions between 10 to 20%. However, a proposed predictive scheduling tool did not lead to a better performance.

In a study focusing on the lifecycle assessment of a smart waste collection in the Swedish municipality Södertälje, Chiew and Brunklaus (2021) found out, that the CO_2-equivalent (CO_2eq) of the smart waste bins in the current system only accounts for about 3% of the total emission, while the use of trash bags accounts for the main share of CO_2eq with 96% and the transportation accounts for just 1% of the total emission, when vehicles drive with biodiesel. In order to offset the environmental impact of the smart waste bin infrastructure, at least 32% of the trash bags need to be reduced. According to Chiew and Brunklaus (2021) the evaluation of the concrete environmental impact of the IoT-based system is challenging due to a lack of precise data.

The use of sensor in waste collection problems is frequently discussed in the literature and its benefits in terms of reducing the number of vehicles and decreasing tour lengths are evident, albeit the environmental impact due to implementing the system itself is still hard to assess. Yet rarely is a sensor allocation in SWM reviewed, which makes use of only allocating a limited number of sensors. Therefore, the sensor placement problem is examined in the following chapter.

2.3.3 The Sensor Placement Problem

When it comes to the sensor placement problem in solid waste collection, literature is scarce. There are several recent articles that investigate how IoT-based solid waste management can be set up. This stream of research is mainly about the sensor types used to sense the filling levels and the IoT-framework that needs to be employed in order to guarantee real-time tracking at a reasonable cost (Akram et al., 2021; Baldo et al., 2021; Cerchecci et al., 2018; Karthik et al., 2021; Ramos et al., 2018; Sohag & Podder, 2020). However, the sensor placement inside a bin and the necessary supporting IoT-infrastructure is not the concern of this thesis. Instead, it focuses on the potential benefit of adding smart sensors to waste bins. The limited allocation of sensors to waste bins, namely which form of sensor allocation is the most suitable, is also of particular interest.

Hemmelmayr et al. (2014) investigate the Waste Bin Allocation and Routing Problem (WBARP), which entails finding a balance between the number of bins allocated to a certain area of collection and the service frequency in which vehicles visit these bins in a given time horizon. In a similar fashion, Roy et al. (2022) extend on this model and also include waste bin penalty costs that occur when waste bins are picked up to late. Their model includes IoT-based smart bins, which show real-time filling levels. Whenever filling levels are at the set threshold of 80%, waste management agencies are notified about the need of collection and bins have to be collected.

While the optimal placement of bins in the system is of importance in the preceding articles, it does not quite fit the sensor placement problem in SWM, in which the bins are already allocated in the system, but the decision is about which bins benefit the most from the added demand information, given the limited number of sensors that can be allocated.

In this regard, Lopes and Ramos (2023) investigate a Sensor-Placement Problem in SWM and develop an algorithm for the allocation of sensors. While their study takes scheduling and routing, as well as sensor information into account, they also use the quality of service and travel efficiency as important measurements. This allows them to not only put forth estimations of distribution, but also produce managerial implications for the operation.

The scheduling presented in Lopes and Ramos (2023) is different to the experimental design in this thesis, as the bins are only collected, when the filling level reaches a previously set threshold. Therefore, the benefit of sensors is more pronounced, as bins can be skipped, whenever the filling level is still below the threshold. In contrast, the scheduling of the customers for each day is already set, and the main benefit of the sensors is that potential overcapacities of the vehicles can be avoided and therefore, returns to the depot can be reduced overall.

In terms of sensor placement positions, the results observed in Lopes and Ramos (2023) indicate that the random placement of sensors performs worst overall. But a higher number of sensors placed in a random fashion achieved better results than a lower number of perfectly placed sensors ($n = 5$ vs. $n = 3$).

Not surprisingly, instances with full information fared better, than with less information. Furthermore, placing sensors only based on the route reduction cost, was only beneficial for instances, where the demand distribution was the same for all locations ($\mu = 20$).

An additional facet that influences the effectiveness of smart waste bins and the allocation of sensors is the demand variability. In this regard, Florio et al. (2022) investigate vehicle routing with stochastic demands and partial reoptimization. The authors compare the policies detour-to-depot, switch, and optimal restocking for differing amounts of load and by using different demand distributions. Their results indicate that a larger demand variability does not lead to bigger improvements, if more refined policies are used. Furthermore, the findings imply higher savings in scenarios with a lower demand variability, allowing for better decision making in regard to restocking or customer switches.

In this chapter several studies have been introduced, which have shown that the use of smart waste bins combined with dynamic routing and scheduling can significantly lower the tour lengths and decrease the number of vehicles used, especially in larger systems and with irregular demands. Additionally, common ways of modelling the stochastic demand in the literature have been shown and the sensor placement problem in the context of IoT-based SWM has been discussed. In regard to sensor allocation of a limited number of sensors in SWM, literature is quite scarce, which is way this thesis can contribute to the body of research.

After investigating variants of the VRP, with a focus on the PVRP with deterministic and stochastic demands in Chapter 2.1, heuristic solutions for solving this problem have been introduced in Chapter 2.2, with a focus on the metaheuristic Tabu Search, which will be used to solve instances of the PVRP with deterministic demand in Chapter 3. Furthermore, the smart waste bin allocation in SWM has been reviewed.

In the following, the solution framework for solving instances of the PVRP with deterministic demands will be presented, which will be compared to results in the literature and serves as the basis for solving the PVRP with stochastic demands and varying levels of waste bins equipped with sensors.

3 Solution Framework for Solving the Periodic Vehicle Routing Problem

Having presented the need for SWM in Chapter 1 and discussed the literature around various forms of vehicle routing problems in Chapter 2 as well as heuristics to solve them, the solution approach to solving the PVRP follows. Starting with Chapter 3.1 the initial solution construction of the PVRP is examined. Subsequently, in Chapter 3.2 the components of the Tabu Search Algorithm used for solving the PVRP instances, established in Cordeau et al. (1997), are described.

3.1 Initial Solution Construction

In order for the Tabu Search to work, an initial solution needs to be constructed. The building process for this is discussed in the following. Since the PVRP is a two-stage problem, which not only consists of solving a VRP, but also entails the assignment of customers to feasible schedules, the initialization stage is not trivial and opens up several options.

The starting solution used for the algorithm is influenced by Russell and Igo (1979), who use the Savings Algorithm established by Clarke and Wright (1964) and adapted it, so that links between customers are only made, if it fits the visiting schedule. To guarantee this, customers are at first allocated to the days in which they are visited. To obtain an initial solution, at first the visiting combinations have been chosen randomly for each customer, as seen in Cordeau et al. (1997). After the visiting schedules have been created, the initial construction of the routes is done using the Clark and Wright Savings Algorithm with euclidean distances between the customers.

Clarke and Wright (1964) introduced this heuristic method for solving the classical Vehicle Routing Problem and it is based on the concept of savings and often produces good initial solutions. As stated by Lysgaard (1997) the Savings Algorithm is used either in a parallel or sequential version. While in the sequential version only one route is built at a time, the parallel version makes use of creating multiple routes at a time and is used in the creation of the algorithm. The general procedure of the savings algorithm is depicted in Table 1.

Table 1

Savings Algorithm Pseudocode

Step 1:	Calculate the savings $s(i, j) = d(D, i) + d(D, j) - d(i, j)$ for every pair (i, j) of demand points
Step 2:	Rank the savings $s(i, j)$ and list them in descending order of magnitude. This creates the "savings list." Process the savings list beginning with the topmost entry in the list (the largest $s(i, j)$).
Step 3:	For the savings $s(i, j)$ under consideration, include link (i, j) in a route if no route constraints will be violated through the inclusion of (i, j) in a route, *and* if: a. *Either*, neither i nor j have already been assigned to a route, in which case a new route is initiated including both i and j. b. *Or*, exactly *one* of the two points *(i or j)* has already been included in an existing route and that point is not interior to that route (a point is interior to a route if it is not adjacent to the depot D in the order of traversal of points), in which case the link (i, j) is added to that same route. c. *Or, both* i and j have already been included in two different existing routes and neither point is interior to its route, in which case the two routes are merged.
Step 4:	If the savings list $s(i, j)$ has not been exhausted, return to Step 3, processing the next entry in the list; otherwise, *stop*: the solution to the VRP consists of the routes created during Step 3. (Any points that have not been assigned to a route during Step 3 must each be served by a vehicle route that begins at the depot D visits the unassigned point and returns to D.)

Note: Reprinted from: (McKelvey & Nguyen, 1983, Chapter 6.4.12)

It is important to mention, that the Savings Algorithm is suitable for Vehicle Routing Problems, in which the vehicle quantity is variable. However, the problem definition of the PVRP has a strict upper limit on the number of vehicles used for each day. For this reason, the Savings Algorithm needs to be adapted for the vehicle number restriction. Hemmelmayr et al. (2009) use this type of adaption for instances in which the number of vehicles in the solution is larger than the allowed number of vehicles. If this is the case, the tour with the minimum number of customers is identified and a customer is removed and inserted in the cheapest position among the remaining tours until the number of routes is feasible, as can be seen in Table 2.

Table 2

Initial solution construction

Apply Clark And Wright
    for each day **do**
        while number of routes > number of vehicles **do**
            shortest_route := find route with fewest number of customers
            for each customer shortest_route **do**
                delete
                insert in cheapest position of the remaining routes
            end for
        end while
    end for

Note: Reprinted from Figure 2 in Hemmelmayr et al. (2009)

This procedure usually works for instances with less restrictive capacity constraints, but this does not always lead to feasible solutions in scenarios, where the capacity constraints are more limiting. The Tabu Search algorithm can deal with starting solutions, where the capacities or time restrictions are not feasible, but the schedule assignments and vehicle numbers are feasible. However, it should be more beneficial to have a starting solution that is feasible, as there are less iterations wasted to reach a feasible state.

This is especially crucial for instances where neighborhoods are large, and computation takes a long time. In these cases, it is important to consider the amount of time as the determinant for the breaking condition rather than the number of iterations. This is because the breaking condition cannot be achieved within a fixed time limit.

Further, it is noteworthy that the Savings Algorithm usually produces decent results, and in some cases not all of the vehicles are used. In events, where the maximum number of vehicles is not exhausted, empty tours are constructed, which are beneficial for the Tabu Search, because the empty tours allow for more flexible moves through their additional capacity. As the use of the vehicles does not incur additional costs in the PVRP, this is detrimental to the outcome.

3.2 Tabu Search Framework

The Tabu Search presented in this chapter is largely influenced by the solution method by Cordeau et al. (1997) and includes some adaptions, where deemed suitable. In Chapter 3.2.1 the objective function and the penalization strategy are explained, followed by the neighborhood structure in 3.2.2. Subsequently, Chapter 3.2.3 introduces the local search in form of the 2-Opt algorithm. Afterwards the Tabu Tenure is discussed in 3.2.4 and the aspiration criteria is described in 3.2.5. Finally, in Chapter 3.2.6 the termination criteria are explained and in Chapter 3.2.7 the Tabu Search pseudocode is discussed.

The underlying Tabu Search Algorithm makes use of a neighborhood structure that permits intermediate infeasible solutions and utilizes a penalization strategy, which enables further exploration of the neighborhood *N(S)*. The Tabu Search algorithm has a specific notation and uses various parameters, which are summarized in Table 3.

Before the Tabu Search Algorithm is explained in detail, all necessary components will be examined at first. The pseudocode will be discussed extensively in Chapter 3.2.7 and is shown in Figure 9. In the following, the functions used to evaluate solutions will be explained at first.

Table 3

Notation used in the description of the Tabu Search algorithm

Abbreviation	Description
B(s)	Attribute set of solution s
c(s)	Routing cost of solution s
d(s)	Excess duration of solution s
f(s)	Cost of solution s
M(s)	A subset of N(s)
g(s)	Penalized cost of solution s
M(s)	A subset of N(s)
N(s)	Neighborhood of solution s
s, $\bar{s}$	Solutions
s*	Best solution identified
α	Penalty factor for overcapacity
β	Penalty factor for overduration
γ	Factor used to adjust the intensity of the diversification
δ	Parameter used to update a and b
η	Total number of iterations to be performed
Ө	Tabu duration
λ	Iteration counter
ρ_{ikl}	Number of times attribute (i, k, l) has been added to the solution
σ_{ikl}	Aspiration level of attribute (i, k, l)
τ_{ikl}	Last iteration for which attribute (i, k, l) is declared tabu

Note: adapted from Table I in Cordeau et al. (1997, p. 108)

3.2.1 Objective Function and Penalization Strategy

One key component of the algorithm is the use of a cost function *f(s)* depicted in equation (11), consisting of not only the routing cost function but also two penalty functions. The first penalty function *q(s)* is used to penalize overcapacity, while the second penalty function *d(s)* penalizes overduration for instances, where time restrictions need to be enforced. Their equations can be seen in (12) and (13), respectively. These two functions have in common, that their respective values are increasing with the amount of overduration/overcapacity and have a value of zero, when there is no overduration/overcapacity. As a result, the *f(s)* values of feasible solutions are only compared by their routing cost, because the penalty terms are both equivalent to 0. By means of the penalization process, the values of infeasible solutions are less attractive, the more capacity or duration restrictions are hurt. For this reason, solutions are selected in the course of the computational run, which violate the restrictions to a decreasingly small extent.

$$f(s) = c(s) + \alpha q(s) + \beta d(s) \tag{11}$$

$$c(s) = \sum_{i=0}^{n}\sum_{j=0}^{n}\sum_{k=1}^{m}\sum_{l=1}^{t} c_{ijkl}\, x_{ijkl} \tag{12}$$

$$q(s) = \sum_{k=1}^{m}\sum_{l=1}^{t} [(\sum_{i=0}^{n}\sum_{j=0}^{n} q_i\, x_{ijkl}) - Q_k]^{+} \tag{13}$$

$$d(s) = \sum_{k=1}^{m}\sum_{l=1}^{t} [(\sum_{i=0}^{n}\sum_{j=0}^{n}(c_{ijkl} + d_i)x_{ijkl}) - D_k]^{+} \tag{14}$$

The factors α and β further influence the results and get updated after every iteration. Current solutions, which are feasible in regard to their capacity, decrease their alpha value by dividing α by (1 + δ), while infeasible solutions multiply the current α by (1 + δ). The beta values are updated correspondingly for current solutions in regard to their duration restriction. The parameter δ is set to 0.5, as this is deemed the most suitable value for these instances, based on a sensitivity analysis by Cordeau et al. (1997).

Solutions are not only compared by their solution value *f(s),* that incorporates the total tour length as well as penalty terms for overcapacity and overduration, but also the additional penalized solution value *g(s)* from Cordeau et al. (1997). In order to achieve a better diversification and as a mean to reduce the likelihood for cycling solutions, the penalized cost of solution *g(s)* incorporates a list of moves, which have been most frequently visited in the current solution. The penalized solution cost *g(s)* is calculated by adding the penalty term to the solution cost. This penalty term consists of the routing costs *c(s)* multiplied by the square root factor $\sqrt{nmt}$, the intensification factor γ (set to 0.015) and the sum of the number of times the new moves have been added to the current solution, divided by the iteration count λ.

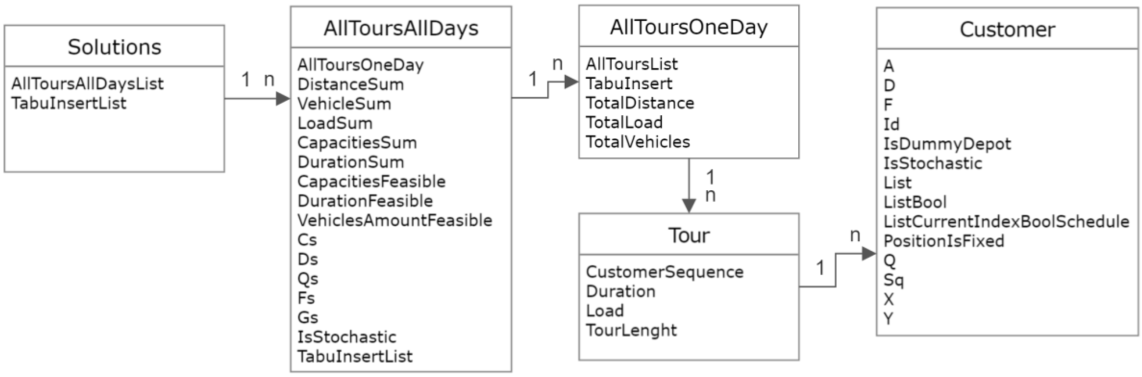
$$g(s) = f(s) + \gamma\sqrt{nmt}c(s) * \sum_{(i,k,l)\in B(\bar{s})\setbackslash B(s)} \rho_{ikl} / \lambda \tag{15}$$

The square root factor is based on research by Taillard (1993) and is used to appropriately account for the solution size, as there are *O (nmt)* attributes in a solution (Cordeau et al., 1997).

Solutions differ from Cordeau et al. (1997) in the regard, that they do not consist of a list of attributes *(i, k, l),* which refer to the customer *i*, that has been moved from tour *k* on day *l*. Instead, solutions consist of a list of days, where each list element consists of a list of tours for the day. Moreover, the tours consist of the visited customers and depots at each end. However, a list of attribute sets *(i, k, l)* is part of every solution (TabuInsertList), that signifies the moves from the current solution. This set will be included in the tabu list, when the solution is chosen as the current solution. The structure of the Tabu Search implementation is briefly summarized in a simplified class diagram in Figure 4.

Figure 4

Simplified class diagram of the Tabu Search implementation



Note: the diagram shows an approximate representation of the program's structure, methods are left out

3.2.2 Neighborhood Structure

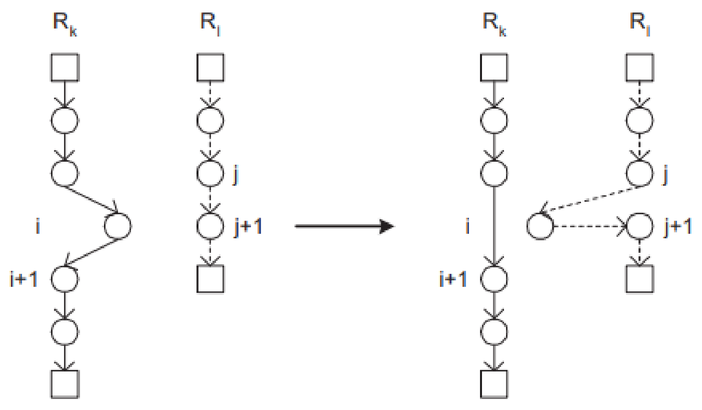
The Neighborhood Structure in the Tabu Search consists of three subsets of neighborhoods. While the Move neighborhood and the Schedule Swap neighborhood have been used in Cordeau et al. (1997), the Customer Swap neighborhood provides an addition to the adapted Tabu Search.

3.2.2.1 Move Neighborhood

The move operator is one of the most basic intra-day inter-tour operators. For every customer $n \in R$, customer n is removed from the tour $R1$ $(R1 \backslash= R2)$ and inserted in a Tour $R2$ of the same day. The complexity of this procedure is $S \mid * \mid R \mid (\mid R \mid -1)$ in the case of the VRP (Cerrone & Sciomachen, 2022). Because the PVRP has a number of days D, the complexity needs to be adapted to $D * (\mid S \mid * \mid R \mid (\mid R \mid -1))$. Figure 5 shows the relocation of one customer from tour $R1$ to tour $R2$.

Figure 5

Relocate operator



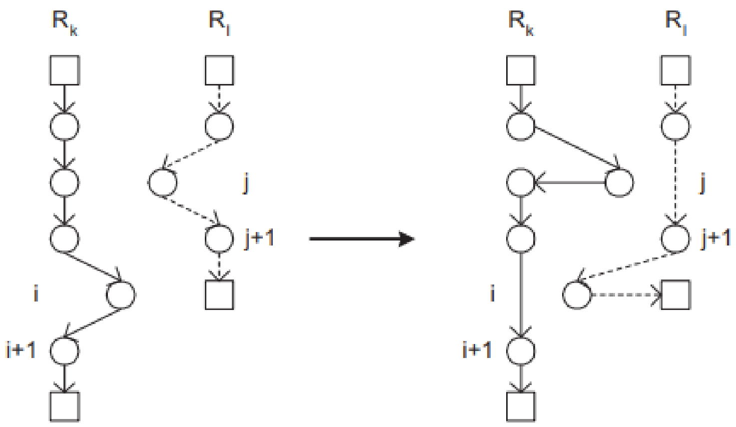
Note: Figure 2 from Ho and Haugland (2004, p. 6)

3.2.2.2 Customer Swap Neighborhood

The customer swap operator removes two customers $n1 \in R1$ and $n2 \in R2$ from two different routes $R1$, $R2 \in S$, removes them and inserts them in the corresponding route, as illustrated in Figure 6. The customer swap operator is used in the customer swap neighborhood in the Tabu Search and makes use of all possible route and customer swap combinations for each day. It is important to note, that no customer schedules are changed, and only customer moves are done. According to Cerrone and Sciomachen (2022) the neighborhood has a complexity of $| S | * (| S | -1) | R1 | R2$, where $R1$ and $R2$ stand for the chosen tours. This has to be adapted for the PVRP, as the number of days accounts for the complexity as well. Therefore, the complexity for the total neighborhood can be summarized as: $D * (| S | * (| S | -1) | R1 | R2)$.

Figure 6

Exchange operator



Note: Figure 3 from Ho and Haugland (2004, p. 7)

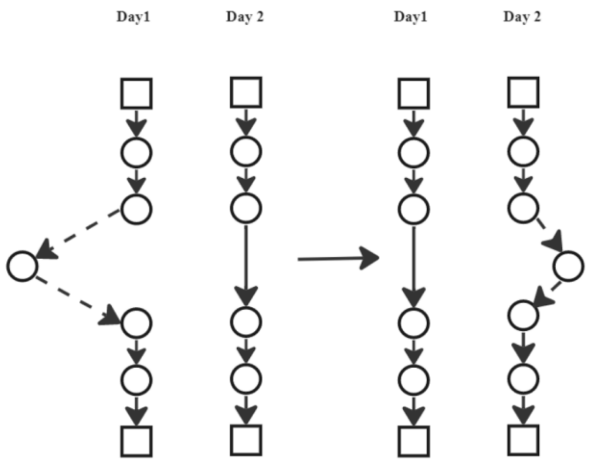
3.2.2.3 Schedule Swap Neighborhood

The schedule swap neighborhood is different to the operators presented before, as it allows to change the days in which the customers are served. At first, for every customer $n \in S$, the customers with non-fixed schedules are returned ($n \in \bar{\bar{S}}$). This leaves out all customers, where no feasible customer swap is possible. Then all possible schedule swaps are returned. For each neighbor, the customer is removed from the day of its current position and inserted in the days, where it needs to be served in regard to its new schedule. The decision of which route R customer n is inserted depends on the value of $f(s)$, which has been explained extensively in Chapter 3.2.1. Therefore, routes are preferred, where not only the routing cost is minimized, but also the amount of overduration and overcapacity is smaller.

In Figure 7 the Schedule Swap Operator is pictured for an example consisting of a two-day time horizon with a single tour. In the example the schedule of the customer is changed, resulting in a deletion of the customer in day 1 and an insertion of the same customer in day 2 at the best possible position.

Figure 7

Schedule Swap Operator



Note: own Illustration

To reduce the risk of a repeating sequence of solutions and to diversify the search, only solutions which do not lead to cycling are used in the neighborhood *M(s)*. The neighborhood *M(s)* consists of all solutions of the previously explained neighborhoods without solutions, which might lead to cycling. For this to work, the solutions are first checked to see if they are tabu. Each solution consists of a list of new moves *(i, j, k)*, which either consists of one move (move neighborhood) or multiple moves (customer-swap neighborhood and schedule-swap neighborhood). For each move, it is checked whether the customer, tour and day index combination is included in the tabu-list. If this is the case for one or more moves, the solution is considered tabu and can only be overwritten if the solution is feasible and *c(s)* of the solution is below the aspiration level of its move attributes. After all solutions in the neighborhood are checked, that is used in the current iteration.

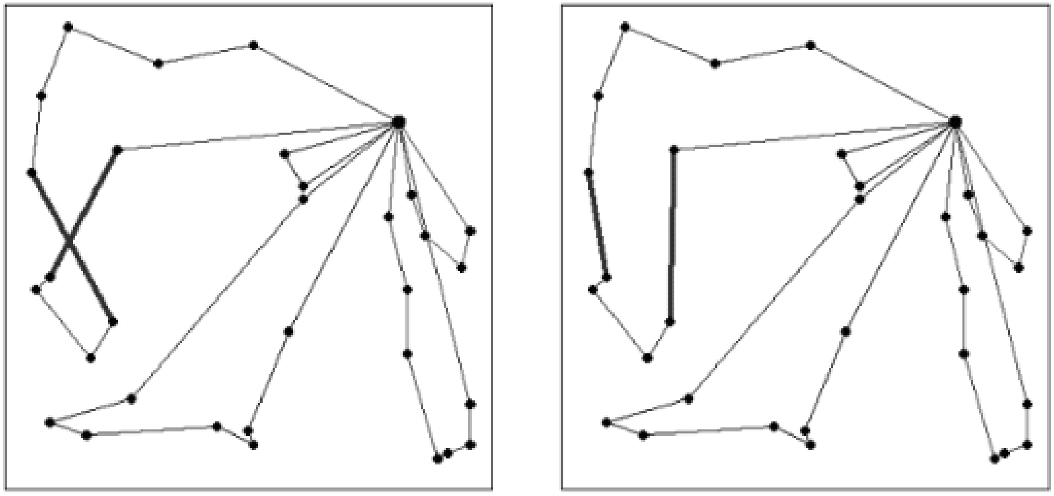
3.2.3 Local Improvement

For the local search in this algorithm, the 2-Opt Intra-Route Operator is used. The 2-Opt move is an operator, which removes two arc *(i, i+1)* and *(j, j+1)* from a tour and replaces it with the pair of arcs *(i, j)* and *(i+1, j+1)* of the same tour. Due to the interchange of the nodes, the sequence from node *i+1* to node *j* is reversed (Doan et al., 2021). For each eligible node combination, the potential saving of a 2-Opt swap is calculated and executed if the swap results in a shorter tour distance. To increase the computational efficiency, a distance matrix for each tour is created, which makes it possible to dispense the calculation of the potential savings at each iteration, as the savings for each possible customer exchange are only computed once. According to Rocki and Suda (2012) the 2-Opt algorithm enables finding local minima in polynomial time with a complexity of O (n^2).

While the 2-Opt algorithm delivers worse results than 3-Opt, the performance difference is substantial as the 3-Opt Local Search has a complexity of O (n^3). Therefore the short computational time can be considered to be one of its main advantages. However, a disadvantage of the 2-Opt method named by Glover and Taillard (1993) is its lack of foresight. According to the authors, crossings between short edges can occur, whenever it is going back from a local minimum. For this reason, using 2-Opt solely as neighborhoods is not a good strategy. A potential improvement of the intra-route 2-Opt heuristic is visualized in Figure 8.

Figure 8

Movement of the 2-Opt Intra-Route Heuristic



Note: Figure 4 from Tavares et al. (2009)

3.2.4 Tabu Tenure

The Tabu List used in this algorithm is based on a constant tabu duration of 7 iterations, as this was proposed by Cordeau et al. (1997) after they conducted a sensitivity analysis to finetune the parameters in the Tabu Search and delivered the best results. In the literature there are examples of using varying tabu durations in order to decrease cycling as suggested in Taillard (1993), but experimenting with this technique did not improve the results. However, the number of solutions in the list fluctuates throughout the computational run because of the varying number of moves of the solution, which depends on its origin of neighborhood. The tabu list consists of a list of moves *(i, j, k)*, that not only give information about the customer, day, and tour index of the move, but also the number of additions of the move to the current solution.

3.2.5 Aspiration Criteria

The aspiration level of a solution is set accordingly to Cordeau et al. (1997) and depends on the current solution. Initially, the aspiration level of a solution is set to *c(s)*, if it is feasible, else the aspiration level is set to ∞. The aspiration level σ_{ikl} of a solution is updated after each iteration and the value is updated to the minimum of either c(s) or σ_{ikl}. As explained in Chapter 3.2.2, feasible solutions with a $c(s)$ smaller than the aspiration level can still be in the neighborhood *M(s),* even if the move of the solution is currently tabu.

3.2.6 Termination Criteria

The termination criteria are quite simple, as the algorithm only stops after 10.000 iterations or whenever 30 minutes have passed for one problem set. Albeit some more complex data sets including many customers, vehicles and days would profit from a longer running time, the time constraint ensured, that one run including the sequential solving of the 42 data sets could be achieved in an appropriate time. Nevertheless, all instances could be solved feasibly and mostly have a total tour length deviation of 5-10% from the original data set.

3.2.7 Tabu Search Implementation

After all components of the Tabu Search have been introduced, the algorithm will be explained in detail. In Figure 9 the general pseudocode for the Tabu Search is depicted. If the starting solution is feasible, the starting solution is set to the best solution, additionally α and β are set to 1. Subsequently, the tabu list is initialized and the duration is set to 7. For the initial solution, the values *c(s), q(s), d(s)* and *f(s)* are computed. Then, starting from line 5, for each iteration the subset of noncycling solutions *M(s)* is set to empty and the neighborhood *N(s)* is generated. For every solution in the neighborhood, it is checked, whether the set of new moves *(i, k, l)* is tabu, or if the aspiration level of the solutions cost *c(s)* is lower than the aspiration level and the solution is feasible. Thereafter, all of the noncycling solutions are saved in *M(s),* and the 2-Opt procedure is used on every solution $\bar{s}$ in *M(s),* but only for the tours, that are affected by the new moves or schedule swaps. The new solutions values are computed and the values of *f(s)* and $f(\bar{s})$ are compared, followed by the computation of the penaltized solution *g(s)*. If the solution is not tabu and *g(s)* smaller than the current solution's *g(s),* then the solution is now the current solution. This also holds true for solutions that are tabu, but its *g(s)* is lower than the best solution found so far. If the solutions *c(s)* is smaller than the best solution's *c(s)*, then it is declared as the current best solution and the aspiration level is updated. Finally, the tabu list and the penalization values are updated. The next iteration follows until the maximum iteration count or the time needed is larger than 20 minutes.

Figure 9

Tabu Search pseudocode

ALGORITHM I: TABU SEARCH

***If s is feasible**, set s* := s. Set $\alpha := 1$ and $\beta := 1$*

TabuList := empty; TabuListDuration := 7

Compute c(s), q(s), d(s) & f(s)

***For** $\lambda = 1, \ldots, \eta$, do*

  Set M(s) := empty

  N(s) := GenerateNeighborhood

  ***For each** $\bar{s} \in N(S)$, do*

    If there exists, (i, k, l) such that $\tau_{ikl} < \lambda$ or such that $\bar{s}$ is feasible and $c(\bar{s}) < \sigma_{ikl}$, set M(s) := M(s) $\cup \{\bar{s}\}$

  ***For each** $\bar{s} \in M(S)$, do*

    TwoOpt for all tours containing each attribute (i, k, l)

    Compute c(s), q(s), d(s) & f(s)

    ***If** $f(\bar{s}) \geq f(s)$, set $g(\bar{s}) := f(\bar{s}) + \sqrt{nmt}\, c(s) \sum_{(i,k,l) \in B(\bar{s}) \setminus B(s)} \rho_{ikl} / \lambda$;else, set $g(\bar{s}) := f(\bar{s})$*

  ***If (!is Tabu) && (g(s') < g(s*))** set s' = $\bar{s}$*

  ***Else if (isTabu) && (g($\bar{s}$) < s*)** set s' = $\bar{s}$*

  ***If** s' is feasible, do*

    ***If c(s') < c(s*)**, set s* := s'*

    ***For each (i,k,l) $\in$ B (s')**, set $\sigma_{ikl} := \min\{\sigma_{ikl}, c(s')\}$*

  Update TabuList

  Update penalty factors

    ***If q(s') = 0**, set $\alpha := \alpha/(1 + \delta)$, **else** set $\alpha := \alpha(1 + \delta)$*

    ***If d(s') = 0**, set $\beta := \beta/(1 + \delta)$, **else** set $\beta := \beta(1 + \delta)$*

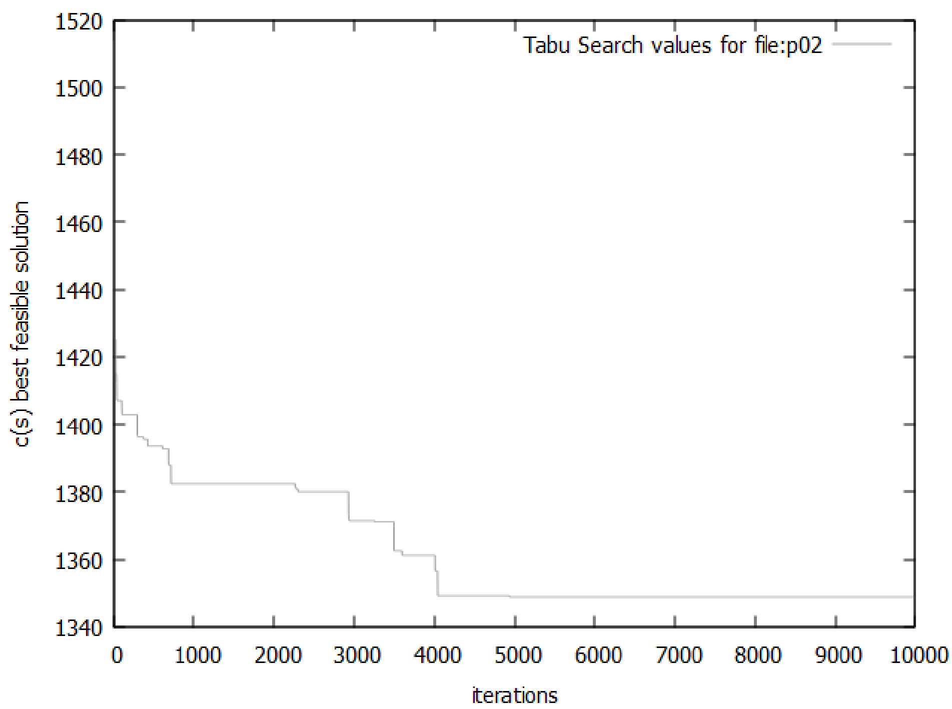
***If** (time elapsed > 20min), break*

Note: based on Cordeau et al. (1997)

With regards to the solution values during the search, the *c(s)* values of the best feasible solution are decreasing, but not necessarily at each iteration. There are plateau-phases, in which no new best feasible solutions emerge for a considerable number of iterations. Through the use of the Tabu List and the frequency-based diversification, local minima can be overcome. In Figure 10 an example of the best feasible solutions during a computational run of the Tabu Search is given.

Figure 10

Best feasible solution per iteration using Tabu Search. problem set p02, deterministic model



Note: best feasible solution value in Cordeau et al. (1997): 1330.09

In this chapter, the general solution procedure for solving the PVRP with deterministic demands has been explained. By using the Savings Algorithm as a starting solution, coupled with a Tabu Search metaheuristic, that has been adapted from Cordeau et al. (1997), all PVRP instances could be feasibly solved. The solutions mostly have a deviation between 5-10% from the results achieved in Cordeau et al. (1997). If a longer computation time would have been used, especially for larger instances, the deviation could have been decreased even more. However, the results form a solid foundation for solving the PVRPSD. In the following chapter, the solution procedure for the necessary adaptions to the PVRPSD and the recourse policies will be explained.

4 Solving the Periodic Vehicle Routing Problem with Stochastic Demands

The solution framework for solving the PVRP with deterministic demands has been explained in the preceding chapter. However, some adjustments have to be done to the framework as the focus shifts from the PVRP with deterministic demands to the PVRPSD. The main design choices for adapting to the new circumstances are presented in Chapter 4.1, while the different recourse actions are presented in Chapter 4.2.

4.1 Adaptions to the PVRPSD

While Chapter 4.1.1 focuses on the variety of options regarding the assignment of sensors, Chapter 4.1.2 gives an understanding about the origin of the stochasticity and the underlying distribution function. Moreover, in Chapter 4.1.3 the concept of fixing visited customers is explained, which is necessary for the reoptimization of routes.

As already discussed in Chapter 2.3.2 Lopes and Ramos (2023) investigate the sensor placement of a bin collection problem. In the article, sensors are placed either based on the sensor value, based on the expected distance reduction or randomly. The random sensor allocation proved to be the worst form of allocation. Their experimental design differs in several aspects from the design in this thesis. First of all, the authors assess several distributions, where the mean of each visited customer is either the same or there are only a few customers with a significantly higher/lower mean demand (μ = 10 vs. μ = 30).

It is important to mention, that the data set with stochastic demand used for this thesis is derived from the PVRP instances with deterministic demands established in Cordeau et al. (1997). Therefore, setting the demands to an arbitrarily chosen level for the PVRPSD data set did not seem appropriate, since this presumably would influence the resulting tours through the demand changes and limit the comparability of the data sets with stochastic and deterministic demands. Secondly, in Lopes and Ramos (2023) the filling levels of waste bins are measured by their percentage of being full. As a consequence, this experimental design also captures overflow cost, which they consider to be a fine for bins that are too full.

Due to the adaption of the data sets from Cordeau et al. (1997) and the use of a distribution function for the demand based on the initial demand values a percentage of the maximum filling level is not suitable. The deterministic demands in the original data set cannot feasibly be set as the maximum filling level as the nature of the distribution function includes values above and below the deterministic demand. For this reason, finding a suitable maximum level would prove to be difficult. Therefore, the bin filling levels will only be considered as absolute values and the design choices for assigning the stochasticity will be discussed in the following.

4.1.1 Design Choices for Assigning Stochasticity

In this chapter the various forms of strategic sensor assignments are discussed at first. While Chapter 2 has dealt with the theoretical background of the PVRP and PVRPSD, the preceding chapter addressed how the PVRP is solved heuristically. However, the instances given for the PVRP need to be adapted for the PVRPSD, thus there is a need to manage the new circumstances of stochastic demands. Whereas the PVRP only covers the case with complete information that occurs when 100% of the waste bins are equipped with sensors, the PVRPSD includes scenarios without sensor-equipped bins or just certain numbers of bins with sensors.

Although the underlying design choice is about choosing the location of where to place the sensors, the method in which this is achieved is actually the reverse. Instead of assigning waste bins with sensors that allow the delivery of deterministic demand information for the routing planning decision, waste bins are chosen to *not* receive sensors, thus delivering only stochastic information about their current demand. Even though this sounds counterintuitive at first, it is a necessary prerequisite, when the instances are based on the previously used instances from Cordeau et al. (1997) with deterministic demands.

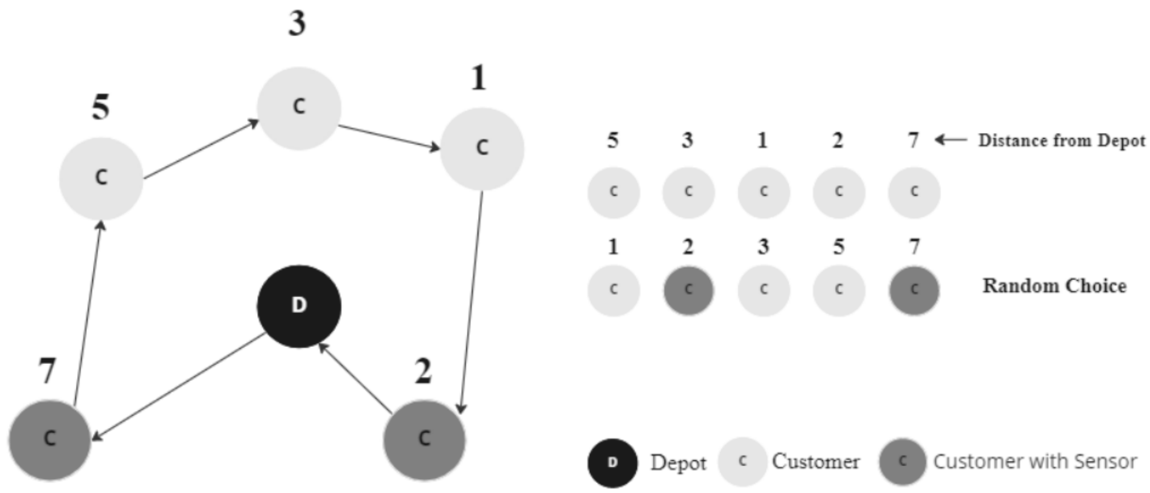
To clarify this further with an example, whenever there is talk of a 40% sensor allocation, 60% of the customers were treated as customers with stochastic demand. As follows, 40% of customers have the same demand derived from the instances, while 60% of customers receive a stochastic demand based on a distribution function. In the following, the sensor assignment choices for the PVRPSD setting are discussed, which serve as basis for comparison of their impact on decreasing the overall tour lengths.

4.1.1.1 Random Sensor Assignment

The random sensor allocation serves as the baseline of comparison to measure the effect of the added demand information. In this case, waste bins are chosen randomly for receiving stochastic demands, respectively the reverse of the waste bins receive sensors and keep their deterministic demands from the original data set. In this method the range of partial sensor allocation is chosen from 20% sensors to 80% sensors in steps of 20. Therefore 20% sensors, 40% sensors, 60% sensors and 80% sensors are used for comparing the results. The case with full information and thus 100% of sensors is already covered in the PVRP case and finally the scenario with 0% sensors is not used, as at least a small percentage of sensors should be used for testing whether sensors are useful to decrease tour lengths. In Figure 11, the random sensor assignment is represented for a single tour with a small number of 5 customers. In this case, 2 out of 5 customers are randomly assigned to receive sensors, which translates to a 40% allocation of sensors or smart bins. These 40% of customers keep the deterministic demands from the PVRP instances established in Cordeau et al. (1997), while the other 60% of customers receive stochastic demands. The procedure for the stochastic demand assignment will be discussed extensively in the following Chapter 4.1.2.

Figure 11

Random Sensor Assignment with 40% sensors

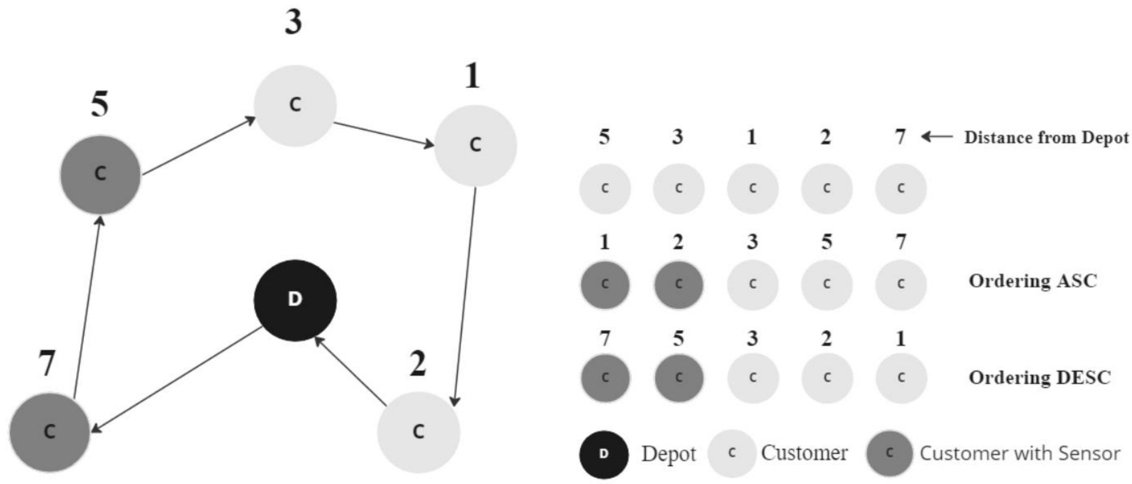


4.1.1.2 Demand-based Sensor Assignment

In the demand-based sensor assignment paradigm, all customers of the given instance are ordered by their demand either in an ascending (ASC) or descending (DESC) manner. After the ordering, depending on the percentage of sensors used for the scenario, the first X% of customers are assigned to be customers with stochastic demands. The rest of the 1-X% customers receive a sensor and keep their deterministic demand of the given instance. In Figure 12 the ordering of customers is represented on the right, while a tour and the allocated sensor-locations are depicted at the left.

Figure 12

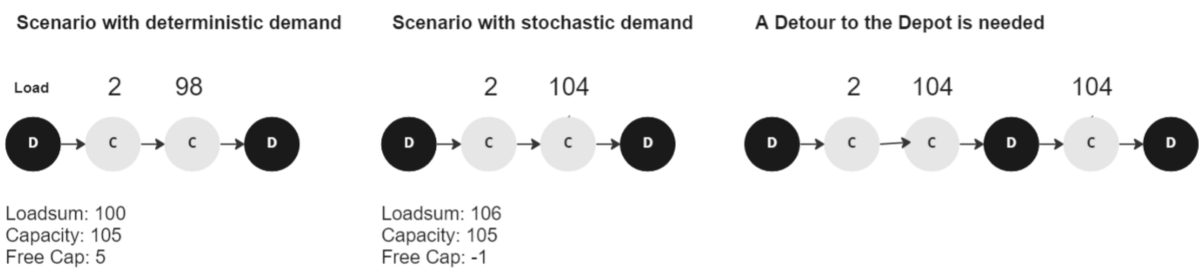
Demand-ordering example with 40% Sensors



To clarify, why the demand-based sensor allocation can be a useful tool in decreasing the overall tour length a small toy example is used in Figure 13, which shows a short tour with only two customers. The tour has a capacity of 105 and the combined load of the customers in the deterministic demand setting is 100, with an uneven distribution of load between the customers (2 vs. 98). In the middle of the figure a potential scenario with stochastic demand is represented, in which the customers are assigned their respective load.

Figure 13

Tour failure example with demand-based ordering

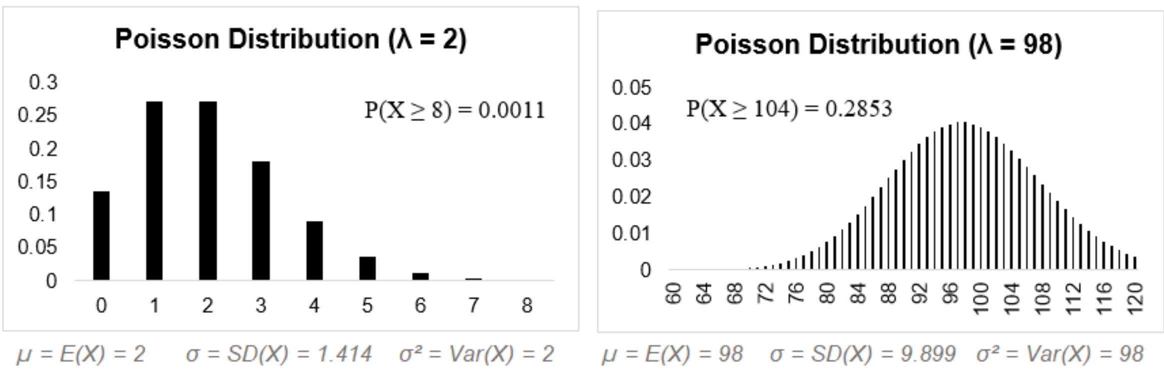


It can be seen that the demand assigned for the first customer stays the same, but the demand for the second customer assigned by the method using the Poisson distribution has risen, so that now there is a capacity overflow, and the load of the customers cannot be collected by the truck anymore, which makes a detour to the depot necessary.

The probability for a failure in the preceding example can be calculated and the values, as well as the distribution function for both of the customers' demand is outlined in Figure 14. The figure illustrates the Poisson distribution of $\lambda = 2$ and $\lambda = 98$, respectively. The probability, that the second customer with demand 98 has a demand of 104 or more in the instance with stochastic demand is *28.53%*, which would result in overcapacity. Meanwhile, the probability, that the first customer with demand 2 has a demand of 8 or more in the instance with stochastic demand, which would result in overcapacity is only *0.11%*. Given this example it can be observed that the failure probability increases, when the customer with high deterministic demand receives a stochastic demand, compared to the customer with a low deterministic demand.

Figure 14

Poisson Distribution functions for different values of λ



Note: influenced by Matt Bognar, 2021

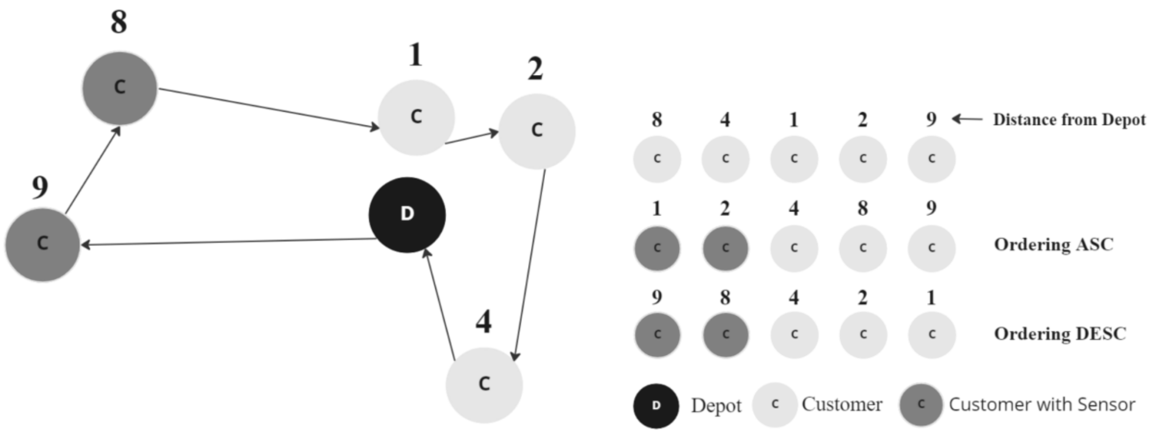
Reversely it should be beneficial to allocate sensors to customers with a higher deterministic demand, so that no stochastic demand will be assigned to it, that has a higher potential to cause overcapacity and lead to detours to the depot.

4.1.1.3 Distance-based Sensor Assignment

Similar to the demand-ordering assignment, in the distance-based sensor allocation paradigm, all customers of the given instance are ordered by their distance to the depot either by ASC or DESC. The assignment of sensors based on the percentage of sensors in the given scenario is conducted in the same fashion and an example for the distance-based sensor assignment is represented in Figure 15. The idea behind the distance-based assignment, is that tour failures should only occur at bins without sensors. Waste bins without sensors should be nearer to the depot, than the waste bins, which are equipped with sensors and are stationed furthest from the depot. Therefore the recourse costs should be lower.

Figure 15

Distance-ordering example with 40% Sensors

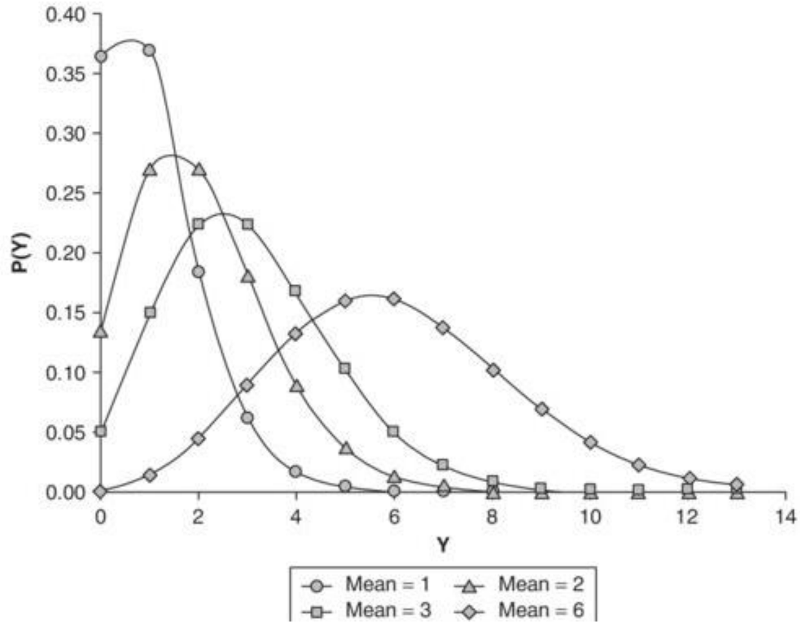


4.1.2 Assigning Stochastic Demand

Depending on the percentage of stochasticity, the chosen percentage of customers is assigned stochastic demand in the order of the list. Once the customers are classified as stochastic, their stochastic demand is calculated. For each stochastic customer a Poisson distribution function is used to return a single Poisson random variable via a method created by Keeler (2019), which is based on the input variable λ. Lambda is equal to the deterministic demand of the customer and is the mean of the simulated Poisson distribution. The Poisson distribution is displayed in Figure 16 for a variety of values for λ.

Figure 16

Poisson Distribution for different values of λ



Note: Figure 21.1 from Nussbaum et al. (2008)

The reason for choosing this function is that only discrete data is returned, which is compatible with the integer demands used throughout the program. The function then returns a stochastic demand *SQ*, that cannot be higher than the total capacity of the vehicle. This counter measurement prevents cycles of returns to the dummy depot and the customer, thus preventing infeasible solutions. For the sake of repeatability and reproducibility, the same seeds for the assignment of stochasticity as well as for the Poisson distribution have been used. It is noticeable, that the distribution function appears to resemble a normal function the higher the value of λ is.

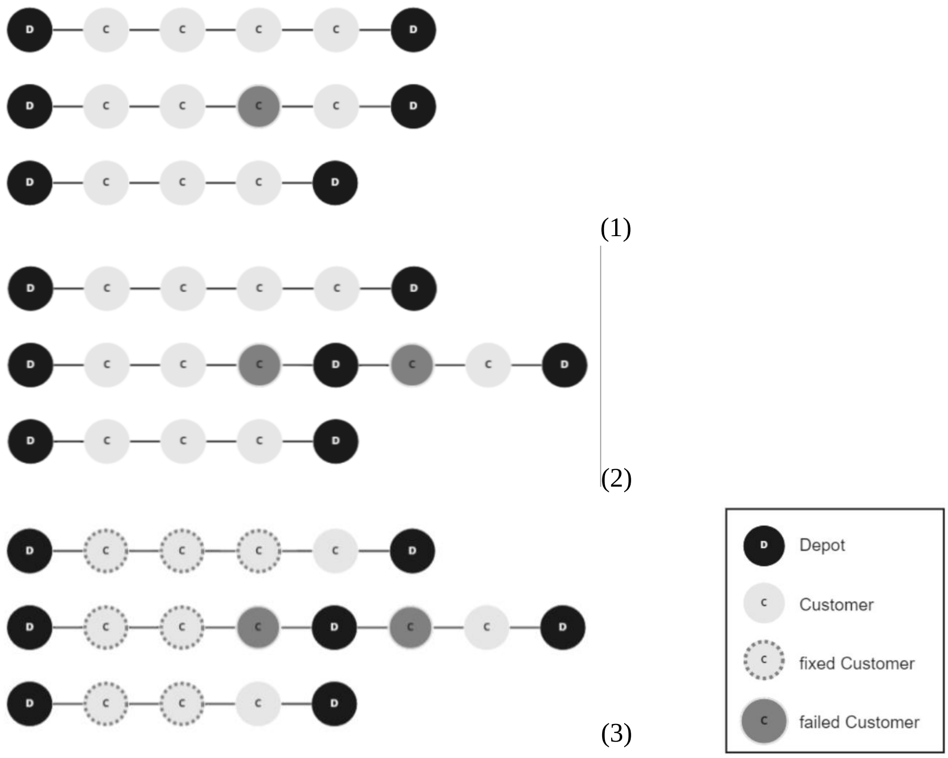
4.1.3 Fixing Customers

Another crucial component needed for adapting the underlying problem to the PVRPSD consists of the concept of fixing customers to tours. It is a necessary prerequisite for adjusting the Tabu Search and ensures that customers, which have already been visited, are not subject of reoptimization. The underlying procedure is demonstrated in the following.

It is important to mention that for the PVRPSD, each day is looked at individually and it is assumed, that every tour starts at the same time at the depot. Furthermore, the speed of the vehicles traveling to the waste bins is assumed to be identical in each of the tours. Customers are visited and for each customer it is checked whether a failure has occurred so far. Depending on the distance between the customers, there might be tours where several customers already have been visited, while the trucks have not yet encountered customers. In the case, of a customer that cannot be served anymore, because the load would lead to overcapacity, a detour to the depot is needed. In the case of *reoptimization after delivery failure*, the distance until failure is recorded and all customers of each tour of the day up until the distance of the customer, where the failures occurred, are considered fixed and cannot be used for reoptimization anymore. In Figure 17, representation of the customer fixing process is visualized. In the first row, a tour depiction is shown, in which the delivery failure occurs at the dark grey customer (1). In the second depiction (2), the detour to the depot has been done after the failed delivery. For this reason, an additional depot is added, and the customer will be visited again. In the bottom row (3), the customers up until the distance of the tour failure are considered fixed and cannot be incorporated in the reoptimization (signified by the outline around the customer).

Figure 17

Fixing customers to tours after delivery failure



4.2 Recourse Policies

For the detours, which are necessary when delivery failures occur, three approaches have been used in the implementation. As the idea behind detours and reoptimization in SVRPs has already been discussed in Chapter 2.1.3, only the necessary adaption to the solution framework will be reviewed. In Chapter 4.2.1 the simplest approach detour-to-depot (DTD) is shown. Afterwards, the reoptimization after delivery failure (RDF) is explained in Chapter 4.2.2 and finally the reoptimization after every customer (REC) is discussed in Chapter 4.2.3.

4.2.1 Detour-To-Depot

In all recourse policies tours are driven according to the deterministic schedule and the actual demand is revealed as the trucks arrive at the customers' location. If a customer cannot be served anymore because the actual demand is higher than the expected demand, the truck has to drive back to the depot, unload and can serve additional customers afterwards. Only in the DTD are routes exactly following the predetermined customer sequence from the solution with deterministic demands. The process of implementation is as follows. In the DTD approach, after the assignment of stochasticity (Chapter 4.1.1) and the assignment of stochastic demand (Chapter 4.1.2), it is checked for each day and for each tour of the day, if the sum of the stochastic demands *SQ* of the tour is larger than the capacity of the vehicle. If this is the case, then a dummy depot is added at the customer, where the tour failure occurred. This dummy depot has a negative stochastic demand equal to the sum of all customers of the tour up until the tour failure. This ensures, that the vehicles capacity of the tour is seen as "empty". After the dummy depot, the customer with the tour failure is visited again and the tour is continued as planned. For instances with time restrictions, the time of the dummy depot is also the negative sum of the accumulated travel time thus far. Customers before the tour failures are seen as fixed (Chapter 4.1.3) and the algorithm stops after all customers are fixed.

4.2.2 Recourse policies with reoptimization

The Reoptimization approaches are recourse policies, which are different from the most simplistic heuristic DTD. As the customer sequence can be altered in the recourse policies with reoptimization, checking the sum of the accumulated *SQs* for each tour previous to the execution is not a good approach and has not been used. Instead, for each day the tours are driven "simultaneously" and reoptimization is done according to the approach. It is assumed, that vehicles can communicate with each other, are driving at the same pace and tours can be adapted flexibly, within the framework of capacity and duration restrictions.

The neighborhood structure in the Tabu Search used for the deterministic version has already been established and consists of moves/customers swaps for each day as well as schedule changes. However, the reoptimization is always based on the tours of the current day, therefore the TS neighborhood is adapted for the new circumstances. Moreover, the neighborhood has been restricted to just intra-day moves, which excludes schedule swaps. The reasoning behind this is, that the predetermined customers schedules should be executed the day they were assigned to. This makes coordination between the drivers much easier and also reduces the computational effort. Due to the scope of the work, optimal restocking is not considered.

4.2.2.1 Reoptimization After Delivery Failure

In the RDF approach, tours are checked for each day simultaneously. The distances driven for each tour are saved and if a failure occurs in any of the tours, the customers of any tour up until the distance of the tour failure are seen as fixed and cannot be used for the reoptimization anymore. After the tour failure, the dummy depot is placed and the reoptimization procedure starts.

The reoptimization procedure is based on the Tabu Search explained in Chapter 3, only a few adaptions have been made. The number of iterations for the Tabu Search are not changed and are kept at 10.000 iterations, while the time for breaking out of the solution is lowered to 10 minutes.

4.2.2.2 Reoptimization After Each Customer

The procedure of REC is quite similar to RDF, however the reoptimization via adapted Tabu Search is done more frequently. All tours are compared simultaneously and the customer with lowest tour distance driven of each tour of the day is fixed and the reoptimization starts. After the reoptimization, the customer with the next lowest driven tour distance is selected for reoptimization and dummy depots are inserted, when it comes to a tour failure. While the reoptimization after each customer should lead to lower overall tour lengths, the time needed for doing Tabu Search after each customer is significantly larger than after just using Tabu Search, whenever a delivery failure is happening. For this reason, the iterations are lowered to 2.000, while the break timer is set to 1 minute.

In this chapter the adaptions needed for solving the PVRPSD have been discussed. Due to the stochastic demands in the problem setting, stochastic demands have to be assigned based on the demand of the instances with deterministic demand. There are several possibilities in choosing which waste bins receive sensors and therefore have deterministic demands, while the rest of the waste bins have stochastic demands. As explained in Chapter 4.1.1, the design choice is based on assigning the stochasticity randomly, based on the demand and based on the distance. Moreover, the stochastic demand is constructed with a Poisson distribution function. When it comes to the reoptimization, the procedure of fixing customers in the tours has been explained to highlight, which customers can be chosen for recourse policies with reoptimization. Finally, in Chapter 4.2 the recourse policies with reoptimization have been explained.

In the following chapter, the computational results will be discussed for the PVRP with deterministic and stochastic demands and the insights will be presented.

5 Computational Results

This section summarizes the computational results achieved for the PVRP instances, as well as the results of the different sensor assignments with varying levels of smart bins. At first, in Chapter 5.1 the input file structure is explained followed by a comparison of the tour visualizations with deterministic and stochastic demands in Chapter 5.2. Then, in Chapter 5.3 the results for the PVRP instances from Cordeau et al. (1997) solved with the Tabu Search are presented. Finally in Chapter 5.4 a comparison of the results for the instances with varying degrees of stochasticity and sensor assignment is given and the results are compared to the instances in the deterministic setting.

For programming the underlying algorithm, the language C# has been used on the Visual Studio IDE using the Community 2022 Edition. The code has been executed on a Windows 11 Yoga 7 14ITL5 laptop with a 11th Gen Intel® Core™ i7-1165G7 @2.80GHz CPU and 16.0 GB of RAM.

5.1 Input Data

The input data is specified in data files, that are described by HEC Montreal (n.d.) and can be seen in Appendix A and Appendix B. Data files are structured in a way, that they give information about the (vehicle routing) *type*, number of vehicles m, number of customers n, and number of days t in the case of the PVRP in the first row of a data set. Subsequently, rows for each day give information about the maximum vehicle load Q and, in case of time restrictions, the maximum duration of a route D is set to a value above 0. Thereafter, the customers and their respective attributes are represented. Each customer has its own *Id*, an x-/ and y-coordinate, a service duration d in case of time restrictions, a demand quantity q, visit frequency f as well as a number of possible visit combinations and the list of the possible visit combination. To represent each visit combination during a specified period, a coding system is used based on binary bit strings and their decimal equivalents. For example, if a customer is visited on days 2 and 4 during a 5-day period, a code is assigned that corresponds to the decimal equivalent of the binary bit string 01010 (HEC Montreal, n.d.). For further clarification, the file structure is shown in Table 5.

Table 4

Data file structure of the PVRP instances

Data File Structure

Type (type)	*Number of Vehicles (m)*	*Number of Customers (n)*	*Number of Days (t)*
1	3	51	2

Duration and Capacity Restrictions

Maximum Duration of Route (D)	*Maximum Load of a Vehicle (Q)*
0	160
0	160

Customer Information

Customer Id (i)	*X-Coordinate (x)*	*Y-Coordinate (y)*	*Service Duration (d)*	*Demand (q)*	*Frequency of Visit (f)*	*Number of Combinations*	*List of possible Visit Combinations*
0	*30*	*40*	*0*	*0*	*0*	*0*	-
Customer Id (i)	*X-Coordinate (x)*	*Y-Coordinate (y)*	*Service Duration (d)*	*Demand (q)*	*Frequency of Visit (f)*	*Number of Combinations*	*List of possible Visit Combinations*
1	*27*	*52*	*0*	*7*	*1*	*2*	*1,2*
2	*49*	*49*	*0*	*30*	*1*	*2*	*1,2*
3	*52*	*64*	*0*	*30*	*1*	*2*	*1,2*

Note: extract of data set 01 from Cordeau PVRP; not all customers are depicted

5.2 Tour Visualization

In order to visualize the routes, the open-source Python tool proute by Dr. Fabien Tricoire is used. The program allows to display several types of vehicle routing problems and can manage the PVRP instances from Cordeau et al. (1997) if the correct file format is used. Two files must be selected for the routes to be displayed. The first pif-file provides information about the customers/nodes, the second psf-file is also called solution file and is needed to display the order of the visited customers in the tour. In Table 5 an example for data set p03 is used and displays the typical structure in a psf-file. The first 5 lines can be seen as a header, which is needed for the program to determine the file structure. Afterwards the solution for each day is shown. The cost (tour length) is shown in line 8, followed by the duration in line 9 and the route that has been used in line 10. After the route in square brackets, the tour length is given at the first position, the second position is always equal to zero and does not serve a purpose for the PVRP, while the load of the vehicle is shown at position 3. At position 4, the number of the tour's day is given. Finally, arcs between the customers are specified starting from line 11. Here, the x-/ and y-coordinates are listed, coupled with the demand at the last position.

Table 5

Proute solution file p03 deterministic

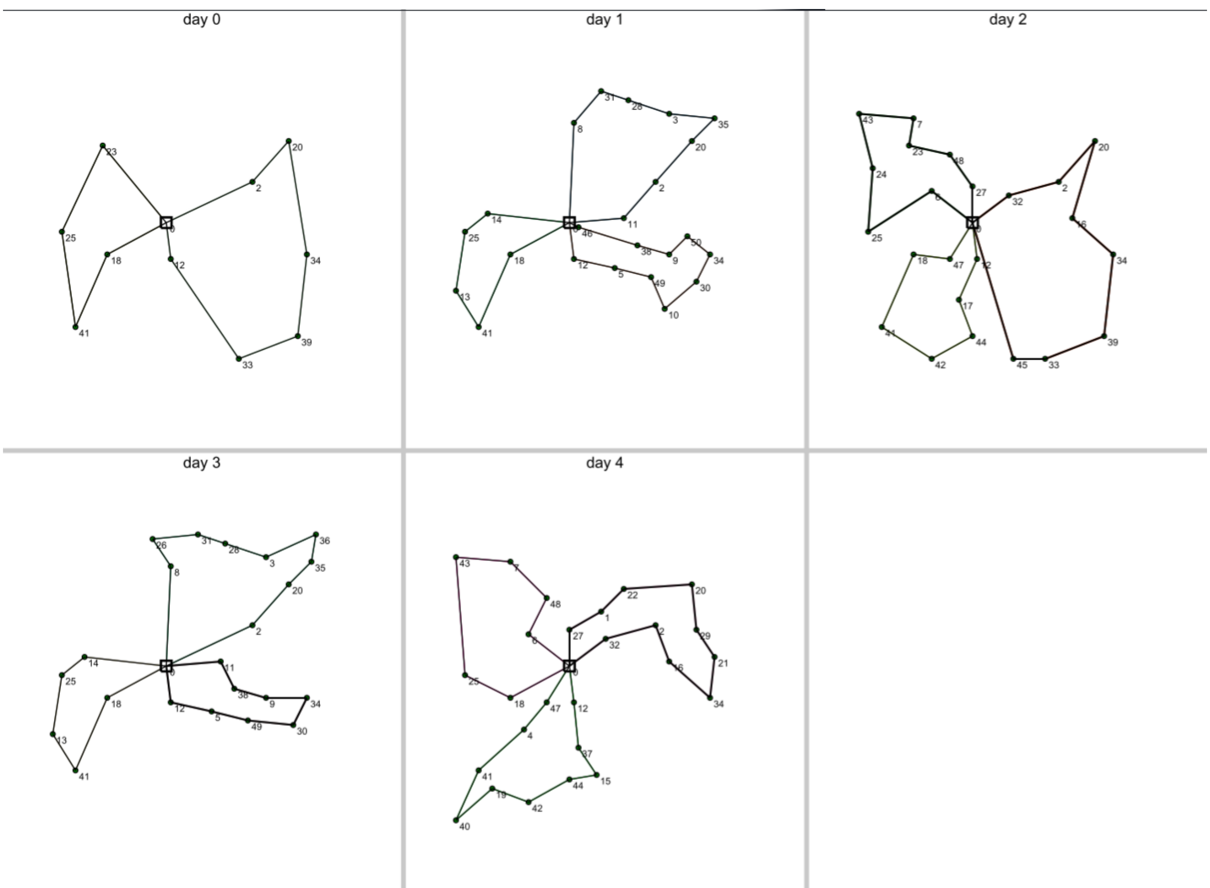
line	file with deterministic demand	file with stochastic demand
1	proute solution file v1;	proute solution file v1;
2	route;node	route;node
3	sequence;cost;duration;load;day	sequence;cost;duration;load;day
4	arc;from;to;load	arc;from;to;load
5	routenode;	routenode;
6	node;index;just before depot	node;index;just before depot
7	#day0	#day0
8	;cost;96	;cost;125
9	;duration;0	;duration;0
10	route;[0, 11, 38, 9, 50, 34, 30, 39, 49, 5, 46, 0];96;0;158;0	route;[0, 11, 38, 9, 50, 34, 30, 39, 49, 5, -1, 5, 46, 0];125;0;179;0
11	arc;0;11;19	arc;0;11;19
12	arc;11;38;15	arc;11;38;15
13	arc;38;9;11	arc;38;9;11
14	arc;9;50;10	arc;9;50;10
15	arc;50;34;26	arc;50;34;26
16	arc;34;30;19	arc;34;30;19
17	arc;30;39;14	arc;30;39;14
18	arc;39;49;18	arc;39;49;18
19	arc;49;5;21	arc;49;5;21
20	arc;5;46;5	***arc;5;-1;0***
21	arc;46;0;0	***arc;-1;5;21***
22		arc;5;46;5
23		arc;46;0;0…

The structure of the solution file for the route visualization in the stochastic setting is almost identical, with the exception of the depots that are used after a delivery failure. With the intention of minimizing any potential interplay of the local search operator, the depot after failure has an ID of -1 instead of the usual ID of 0 for depots. In the same table, the solution file structure for the same data set with stochastic demands is represented in the right column.

A typical result for a solved instance is presented in Figure 18, in which the tours in the deterministic setting of instance p02 are shown. In this visualization, the tours are grouped for each day for a better viewing experience.

Figure 18

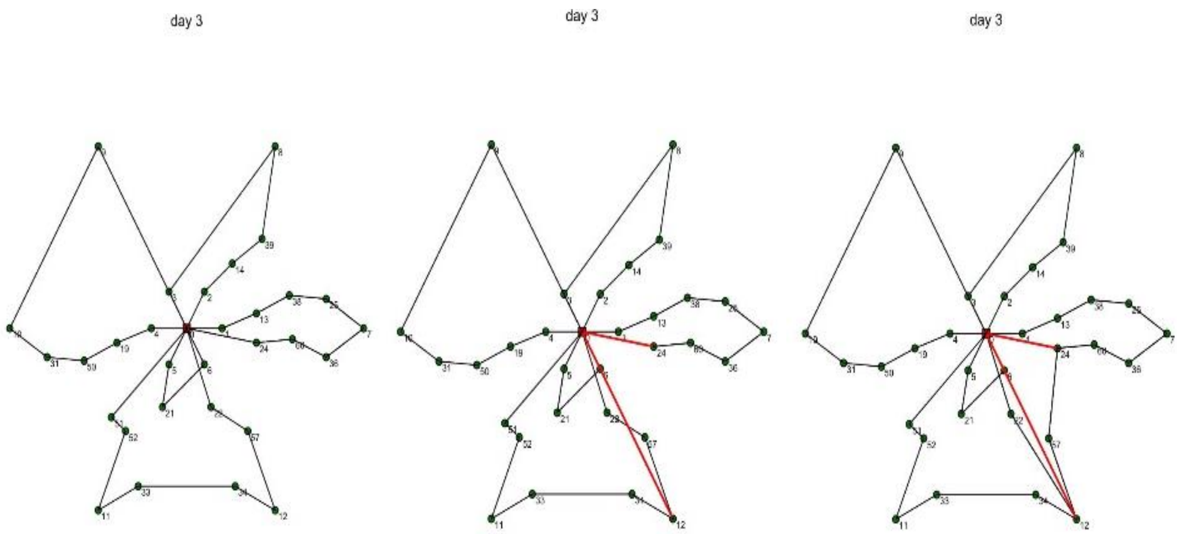
Tour visualization for data set p02; deterministic setting



In the second step of the visualization process, not only the tours in the deterministic setting are shown, but they are compared to the tours in the stochastic setting with the DTD recourse policy, as well as the reoptimization policies. In Figure 19, the different settings are shown for day 3 of instance p21 and a stochasticity level of 80% with random sensor allocation. It is apparent, that the tours in the deterministic setting are almost identical to the DTD setting, only at the position of tour failures a return trip to the depot is done. In the reoptimization after failure approach, tours are reoptimized after a delivery failure, which results in slightly adapted tours.

Figure 19

Route visualization comparison between deterministic and stochastic setting with/without Reoptimization



Note: LEFT: p21 deterministic solution, day 3
MIDDLE: p21, 80% stochasticity, day 3, DTD
RIGHT: p21, 80% stochasticity, day 3, TS Reoptimization

As a consequence of the detours, tour lengths in the setting with stochastic demands are longer. However, the reoptimization procedure could reduce the tour lengths in this specific instance as can be seen in Table 6. While the total tour length is 11,64% longer in the stochastic setting with DTD as recourse policy, the reoptimization after delivery failure approach could reduce the tour length deviation to 9.89%.

Table 6

Tour Length Comparison in varying Scenarios

Scenario	**Stochasticity**	**Tour Length**	**Tour length Δ**
deterministic	0%	2276.86	-
stochastic DTD	80%	2576.85	11.64%
stochastic TS Reopt After Failure	80%	2526.74	9.89%

Note: all scenarios are based on instance p21

In order to get a more comprehensive picture about the results of the difference between deterministic and stochastic scenarios, Chapter 5.3 will provide an overview about the results achieved with the Tabu Search on all instances.

5.3 Results for the Instances with Deterministic Demand

The results for the PVRP instances solved by Cordeau et al. (1997) are presented in Table 7. Regarding the origin of the instances 1-32 represented in Table 7, they are ordered as following. Data sets 1-10 are from Eilon et al. (1974), while Instance 11 is from Russell and Igo (1979). Instance 12 is provided by Cook and Russell (1978), whereas instance 13 is from Russell and Gribbin (1991). The instances ranging from 14 to 32 were established by Chao et al. (1995).

In Table 7, the instances are also classified by their corresponding attributes namely customer amount n, vehicle amount m, number of days t, duration restriction D and the capacity restriction Q. It is important to highlight that the instances are heterogenous in regard to their structure, as their attribute values are varying, which can be seen exemplary in the number of customers ranging from 50 to 413 or the differing time horizons ranging from 2 to 10.

Table 7

Comparison of own results with solutions in Cordeau et al. (1997)

I	n	m	t	D	Q	c(s) CGW	c(s) DK	T CGW	T DK	Iterations	delta	% deviation
1	50	3	2	-	160	524.61	642.23	3.39	7.31	10000	117.62	22.42%
2	50	3	5	-	160	1330.09	1384.47	4.06	13.36	10000	54.38	4.09%
3	50	1	5	-	160	524.61	600.07	3.73	5.54	10000	75.46	14.38%
4	75	6	5	-	140	837.94	909.27	5.19	15.04	10000	71.33	8.51%
5	75	1	10	-	140	2061.36	2133.71	7.48	20.02	4742	72.35	3.51%
6	75	1	10	-	140	840.30	1048	7.84	13.55	10000	207.70	24.72%
7	100	4	2	-	200	829.37	908.99	7.63	20.02	5332	79.62	9.60%
8	100	5	5	-	200	2054.90	2203.1	10.7	20.04	2674	148.20	7.21%
9	100	1	8	-	200	829.45	875.66	10.03	20.01	5493	46.21	5.57%
10	100	4	5	-	200	1629.96	1887.08	9.68	20.03	3291	257.12	15.77%
11	126	4	5	-	235	817.56	862.59	14.17	20.04	2120	45.03	5.51%
12	163	3	5	-	140	1239.58	1399.81	18.37	20.04	1519	160.23	12.93%
13	417	9	7	-	2000	3602.76	4314.11	59.98	20.23	607	711.35	19.74%
14	20	2	4	-	20	954.81	962.6	1.15	1.01	10000	7.79	0.82%
15	38	2	4	-	30	1862.63	1917.73	2.58	5.55	10000	55.10	2.96%
16	56	2	4	-	40	2875.24	3053.94	4.28	19.19	10000	178.70	6.22%
17	40	4	4	-	20	1597.75	1662.74	3.01	7.34	10000	64.99	4.07%
18	76	4	4	-	30	3159.22	3259.19	6.46	20.02	5743	99.97	3.16%
19	112	4	4	-	40	4902.64	5067.97	11.9	20.06	2011	165.33	3.37%
20	184	4	4	-	60	8367.40	8999.39	23.44	20.24	408	631.99	7.55%
21	60	6	4	-	20	2184.04	2276.02	5.2	20.01	8418	91.98	4.21%
22	114	6	4	-	30	4307.19	4474.46	11.46	20.08	1836	167.27	3.88%
23	168	6	4	-	40	6620.50	6894.66	19.58	20.20	768	274.16	4.14%
24	51	3	6	-	20	3704.11	3866.1	4.26	9.34	10000	161.99	4.37%
25	51	3	6	-	20	3781.38	3884.69	4.34	9.38	10000	103.31	2.73%
26	51	3	6	-	20	3795.32	3857.8	4.26	9.09	10000	62.48	1.65%
27	102	6	6	-	20	23017.45	24096.39	11.31	20.02	3843	1078.94	4.69%
28	102	6	6	-	20	22569.40	24274.36	11.13	20.02	3533	1704.96	7.55%
29	102	6	6	-	20	24012.92	24865.33	11.22	20.02	3674	852.41	3.55%
30	153	9	6	-	20	77179.33	84480.2	20.72	20.08	1691	7300.87	9.46%
31	153	9	6	-	20	79382.35	84182.88	20.3	20.09	1604	4800.53	6.05%
32	153	9	6	-	20	80908.95	88277.75	20.62	20.08	1546	7368.80	9.11%
a	48	2	4	500	200	2234.23	2264.82	3.86	9.39	10000	30.59	1.37%
b	96	4	4	480	195	3836.49	3878.08	9.79	20.05	4074	41.59	1.08%
c	144	6	4	460	190	5277.62	5544.22	18.62	20.14	1534	266.60	5.05%
d	192	8	4	440	185	6072.67	6375.63	28.24	20.34	854	302.96	4.99%
e	240	10	4	420	180	6769.80	7104.25	36.62	21.01	655	334.45	4.94%
f	288	12	4	400	175	8462.37	8714.99	50.97	21.04	333	252.62	2.99%
g	72	3	6	500	200	5000.90	5100.17	9.63	20.03	5059	99.27	1.99%
h	144	6	6	475	190	7183.39	7465.97	24.05	20.22	1162	282.58	3.93%
i	216	9	6	450	180	10507.34	10815.84	45.65	21.02	460	308.50	2.94%
j	288	12	6	425	170	13629.25	13572.32	71.56	21.00	268	-56.93	-0.42%

Therefore, it comes as no surprise, that the computational time needed to reach the set amount of 10.000 iterations for each instance varies significantly. While instance 1 was solved by the Tabu Search algorithm in Cordeau et al. (1997) in under 4 minutes (as can be seen in row *T CGW*), it took the algorithm approximately 60 minutes to solve instance 13. Another key point that needs to be pointed out, is that the values of the Tabu Search in this thesis are compared with the values in the row *c(s) CGW*.

In contrast, the instances *a-j* do have a set duration restriction. Similar to the instances previously mentioned, their attribute values are not homogenous, resulting in varying run times.

In the following, the values for each of the PVRP instances from Cordeau et al. (1997) are compared with the results achieved by the adapted Tabu Search Algorithm constructed for this master's thesis. In Table 7, this comparison is also depicted. For each instance, the total tour length of Cordeau et al. (1997) is represented in the second column *c(s) CGW*, while the results with the Tabu Search constructed for this thesis are shown in the third column *c(s) DK*. Simultaneously, the computing time for each iteration is shown in column *T CGW* and *T DK*.

One noticeable distinction regarding the time needed for solving the instances is the cutoff at about 20 minutes for some instances in the Tabu Search variant created for this thesis. As already explained in Chapter 3.2, this was done in order to generate results in a timely manner. For the instances where the time limit is reached, the iteration limit of 10.000 could not be exhausted by the algorithm. The absolute tour length deviation from the results in Cordeau et al. (1997) is shown in column *delta*, while the relative deviation is shown in column *% deviation*. When the results of the PVRP instances in the deterministic setting are analyzed, a few things have to be pointed out.

It has to be emphasized that the results of all instances have been achieved in one run and they do not necessarily represent the best solutions achieved for each instance. Due to the random assignment of customer schedules in the beginning of the algorithm, the results are greatly influenced by the starting decision. The initial solutions of all PVRP instances in the deterministic setting are saved as a file that can easily be read in. The seeds and parameters used are shown in Table 8. For the assignment of stochasticity *randomSeedStochAssignment* has been used, while for the assignment of the Poisson Distribution the seed *randomSeedPoissonAssignment* was utilized.

Table 8

Parameters and seeds used for the solution

parameter name	**value**
randomSeedStochAssignment	new Random(4)
randomSeedPoissonAssignment	new Random(4)

Note: the original seed value for the initial deterministic starting solutions was set to Random().
The solutions for the PVRP instances in the deterministic setting can be read in with a method and were used for all following stochastic settings

There are some instances for example instance 2, instance 6 and instance 13 that proved to be harder to solve, thus resulting in much worse results than in the other problem sets. In the initial stages of the creation of the algorithm, where only feasible solutions were used for the initialization, only rarely feasible solutions could be created.

Moreover, some instances with large neighborhoods could definitely profit from longer runtimes, as only a few hundred iterations could be achieved in the set time limit.

Another noteworthy aspect is that the instances a-j with time restriction were solved with a relatively small tour length deviation. These instances were randomly generated by Cordeau et al. (1997) and designed as a cluster of customers around several points.

When it comes to the performance of the algorithm in the deterministic setting, the percentage of the tour length deviation from the original values is an important measure. Overall, the percentage of solutions with a tour lengths deviation below 10% and below 5% is shown in Table 9. While 86% of solutions are below the threshold of 10% tour length deviation, only 57% have less than 5% deviation from the tour lengths in the original solutions.

Table 9

Percentage of own solutions with deviation below given threshold

Threshold	Abs. number of instances	% of instances
10%	36	86%
5%	24	57%

Note: based on the 42 PVRP instances from Cordeau et al. (1997)

While these results are not particularly close to the performance of well-established algorithms for solving the PVRP in the literature, they still highlight the potential of the Tabu Search Algorithm, as all of the instances could be solved feasibly. With a little fine-tuning through performance optimization, a more sophisticated local search algorithm and a longer run-time, better results could be achieved. Nevertheless, the results of the PVRP instances in the deterministic demand setting can be used as a baseline to compare the results for the instances with stochastic demand and allow the evaluation of different forms of sensor allocation for IoT-based SWM.

5.4 Comparison of the Results with Varying Degrees of Stochastic Demand

In this section the results of the instances with varying percentages of sensors/stochastic customers are compared with the results in the original data set, in which all demands are purely deterministic. The comparison considers two types of recourse policies. While the recourse policy without reoptimization uses DTD as a course of action, the recourse policy with reoptimization utilizes the adapted Tabu Search Algorithm without schedule swaps explained in Chapter 3.2**.** At first the results are compared using random sensor allocation, with varying percentages of sensors, which acts as a baseline for estimating the potential of adding gradually more information about the filling levels compared to a purely stochastic setting.

5.4.1 Results for the Random Sensor Assignment

In Figure 20, a graphical representation of the average tour length deviation between the scenario with deterministic demands and scenarios with stochastic demands is illustrated for 5 levels of sensor percentages (0-20-40-60-80%). The scenario with 100% sensor information is not illustrated, as it is identical to the fully static and deterministic setting, consequently the deviation would be 0%. The results of the random sensor allocation indicate, that with an increase in sensor information, the average tour length deviation decreases and resembles an inverse linear relationship, with a slight increase in the tour deviation from 40% to 60% sensor allocation. The results for each instance are shown in Appendix C and Appendix D.

Figure 20

Average tour length deviation between instances with deterministic demands and instances with varying levels of stochastic demand and random sensor allocation

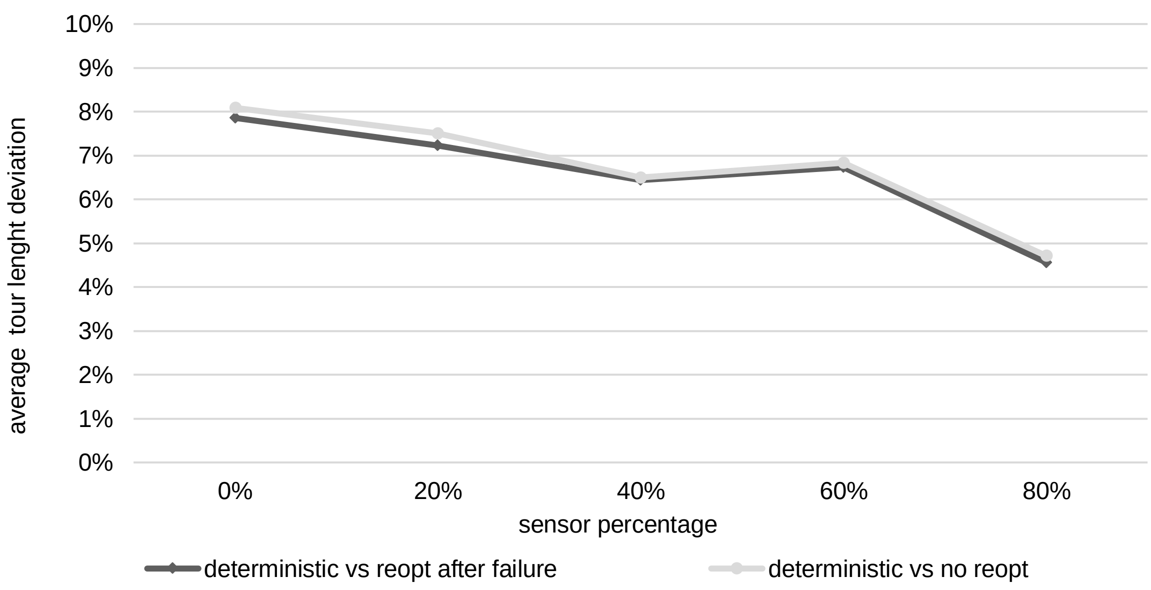


Table 10 shows the numerical comparison of the settings with and without reoptimization after tour failures. It is noteworthy, that the recourse policy using the adapted Tabu Search and intra-tour exchanges after delivery failures performs better in all settings and reduces the average tour length deviation between 0.10% and 0.28% depending on the percentage of smart bins.

Table 10

Numerical comparison of average tour length deviation between instances with deterministic demands and instances with varying levels of stochastic demand

Waste bins equipped with sensors in %	0	20	40	60	80
Average deviation between instances with deterministic demands vs. stochastic instances without reoptimization	8.09%	7.51%	6.50%	6.84%	4.71%
Average deviation between instances with deterministic demands vs. stochastic instances with reoptimization after failure	7.86%	7.23%	6.45%	6.74%	4.56%
Improvement over DTD recourse policy	0.23%	0.28%	0.05%	0.10%	0.15%

Note: the average tour length comparisons are based on the original instances from Cordeau et al. (1997) and the modified instances using varying percentages of customers with stochastic demand in which the demands are assigned based on a Poisson distribution using the original demands as input. The % deviation is based on the average deviation across all 42 instances.

Results of the advanced Sensor Allocation

To evaluate whether a more sophisticated allocation of sensors has a stronger effect on reducing the average tour lengths compared to the scenario with random sensor allocation, two forms of sensor allocation namely distance-based sensor allocation and demand-based sensor allocation are compared, which have been described in Chapter 4.1.1.

5.4.2 Results for the Demand-Based Sensor Assignment

The idea behind the demand-based sensor allocation is to assign sensors to waste bins with high filling levels, as they have a larger probability for a higher absolute deviation from their estimated filling level. Consequently, their potential for resulting in delivery failures is more severe than for waste bins with average or below average filling levels.

It is important to mention that this technique makes use of the demand levels in the instances by Cordeau et al. (1997), in which demands are not on the same level, but are different from customer to customer. In the case of the instances a-j generated by Cordeau et al. (1997), the customer’s demand is modelled via a discrete uniform distribution between [1, 25].
This allows to capitalize on the varying demand levels, yet it has to be noted, that in a real-world situation, demand/filling levels might not diverge as much and are often correlated.

For the purpose of showing whether assigning sensors based on the filling levels of waste bins has an impact on the average tour lengths, two scenarios are created. In the first scenario depicted in Figure 21, smaller percentages of sensors (10-20-30-40%) are assigned to waste bins with high filling levels whereas in the second scenario, shown in Figure 22, a counterexample is used in which the sensors are assigned to customers with the lowest demands.

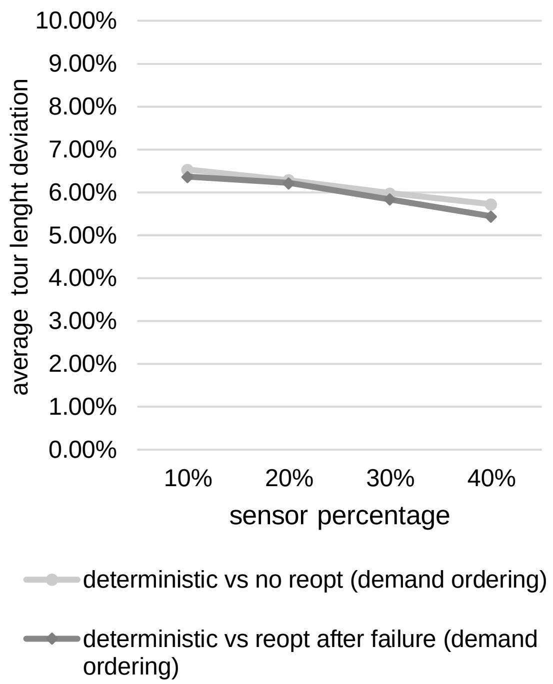
The decision for using only a limited amount of up to 40% sensor was made for several reasons. For one, implementing and maintaining a IoT infrastructure is costly, thus a smaller scale cheaper solution might already prove to be beneficial for reducing tour lengths. Secondly, for the distance-based and demand-based setting, there are two versions each for allocating sensors to the bins, which potentially benefit the most/least from the sensors (ASC vs DESC ordering), thus with less than 50% smart bins overlaps of assigning sensors to the same bins are avoided.

In Figure 21, the average tour lengths deviation increases with a decreasing level of sensors almost linearly, similar to the setting with random sensor allocation. However, for the same number of sensors the average tour length deviation is at a lower level and the slope is more gradual.

In Figure 22, the results appear more ambiguous, as the linear relationship between the level of sensors and the average tour length deviation is less pronounced. It is noteworthy, that while the deviation increases from 40% sensors to 30% sensors as expected from this point on the deviation decreases even though the number of sensors decreases as well.

Figure 21

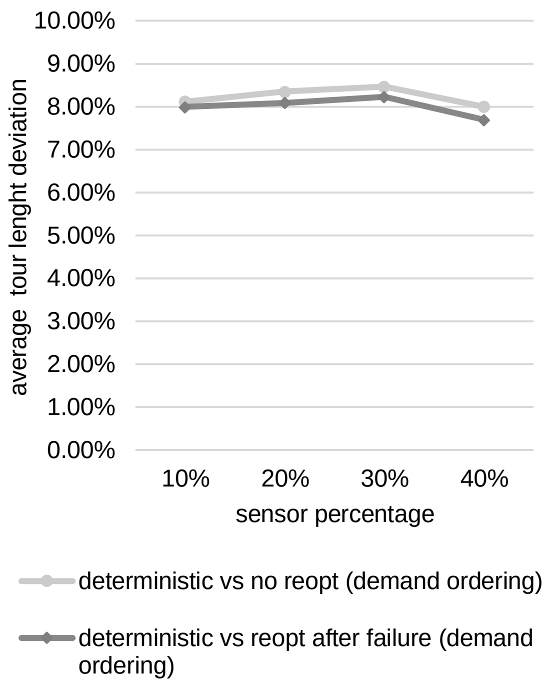
Average tour length deviation between all instances with deterministic demands and instances with varying levels of stochastic demand and demand-based sensor allocation (ASC ordering)



Note: waste bins with high load are chosen first to receive sensors

Figure 22

Average tour length deviation between all instances with deterministic demands and instances with varying levels of stochastic demand and demand-based sensor allocation (DESC ordering)



Note: waste bins with low load are chosen first to receive sensors

When both sensor assignments are compared, it is noticeable that for each level of sensors, the average tour length deviation is lower in the scenario in which the bins with high filling levels are outfitted with sensors. This is especially visible when the sensor levels are at a higher level, but even for scenarios with only 10% sensors the difference between both forms of allocations are noticeable. In Appendix E-H the values for each instance of demand-based sensor assignment with ASC and DESC ordering and varying sensor levels are summarized.

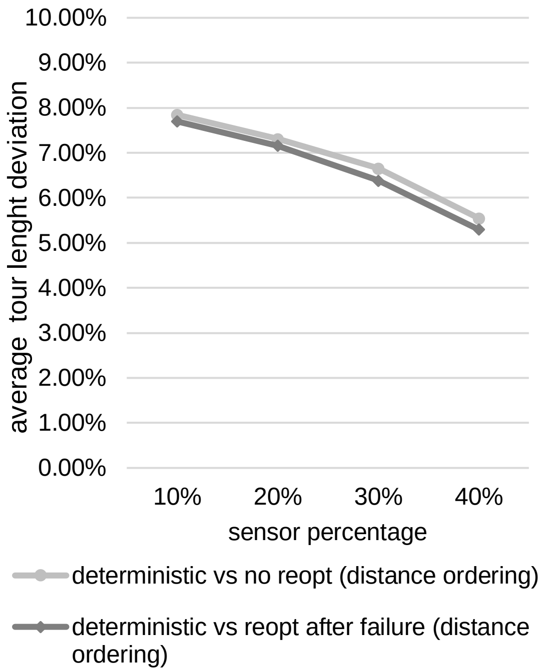
5.4.3 Results for the Distance-Based Sensor Assignment

The second advanced sensor allocation procedure is based on the distance of the bins to the depot and is rooted on the idea of reducing the uncertainty of filling levels of bins, which are far from the depot thus delivery failures at these bins would lead to especially long detours- to the depot. The process is discussed extensively in Chapter 4.1.1.3. Similar to the demand-based sensor assignment in the preceding chapter, sensors are not only assigned to bins which are particularly far from the depot, but a counterexample is also given in which bins are equipped with sensors, which are near to the depot. Regarding the number of sensors, the same levels (10-20-30-40%) are used to compare the results.

Just as in the first setting, two scenarios are constructed in which sensors are allocated to waste bins in and ascending and descending manner. Figure 23 depicts the scenario in which the bins furthest to the depot receive sensors, while Figure 24 shows the setting with sensor allocated to the nearest waste bins from the depot.

Figure 23

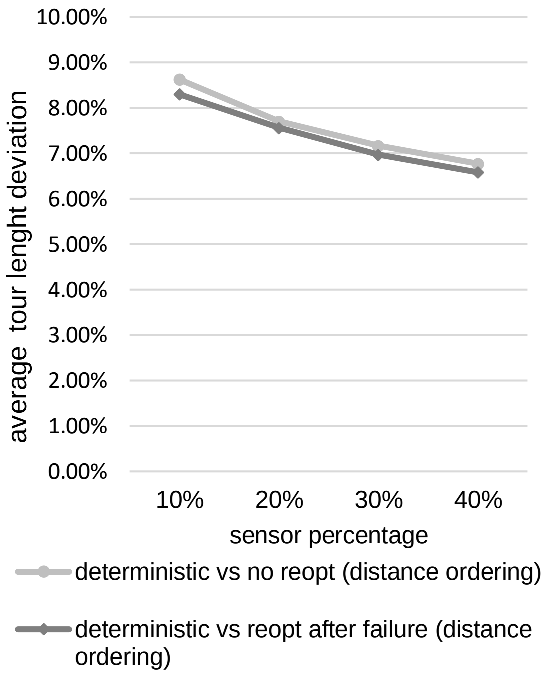
Average tour length deviation between instances with deterministic demands and instances with varying levels of stochastic demand and distance-based sensor allocation (ASC ordering)



Note: waste bins with furthest distance to depot are chosen first to receive sensors

Figure 24

Average tour length deviation between instances with deterministic demands and instances with varying levels of stochastic demand and distance-based sensor allocation (DESC ordering)



Note: waste bins nearest to the depot are chosen first to receive sensors

The graphs show that the average tour length deviation across all instances is apparently much higher when the smart bins are set up nearest to the depot compared to the scenario with smart bin furthest from the depot. This holds true for every level of sensor allocation.

Moreover, the reoptimization after failure performed better in every setting, just like in the random sensor allocation. Additionally, the scenario depicted in Figure 24 performed worse than the scenario with random sensor allocation. In Appendix I-L the values for each instance of distance-based sensor assignment with ASC and DESC ordering and varying sensor levels are summarized.

5.4.4 Results for the Reoptimization after Each Customer

As in the settings before, the sensor levels range from 10-40% and the seed for the initial tours and their demand distributions are identical. In the last case, the random sensor-assignment is used again for the smart bin allocation, what has changed is the use of the reoptimization approach. Instead of comparing the recourse policy DTD with RDF, the Reoptimization procedure is done after every single customer. As one can envision, the computational effort is much larger, as delivery failures are encountered only occasionally and in the case of the REC, the adapted Tabu Search heuristic has to be used for the neighborhood of all unfixed customers after each visited waste bin.

The question is whether the computational effort is worth the potential tour length savings. Due to amount of computation required, the Tabu Search duration and iteration count have been drastically limited. Instead of the maximum iteration of 10.000, only 2.000 iterations are used, and the time limit has been restricted to only 12.000ms for each TS after a customer visit. Still, the computation time needed to generate solutions for all instances of only the scenario with 10% sensors was 30.29h as can be seen in Table 11.

Despite the long total run-time of 30.29h, the results do not look favourable for the REC approach, at least with random sensor assignment. Still, the time for each individual run is simply too short and it is likely that because of it, the Tabu Search cannot lead to any significant improvements as the neighborhoods are only searched insufficiently. A possible other assumption could be that this approach could be considered short-sighted, as improving moves do not take the potential of larger than expected waste bin load into account, as they have not yet been encountered. This could then lead to potentially more failures.

Table 11

Values for all stochastic Instances; 90% Stochasticity; REC

Data Set	Distance	Vehicle Sum	LoadSum	Time (hh:mm:ss:ms)
p1	597.54	6	937	00:00:36.39
p2	1631.01	15	1324	00:01:16.31
p3	692.42	5	636	00:00:00.02
p4	1021.92	10	1065	00:06:23.32
p5	2417.72	28	2621	00:11:18.38
p6	1040.83	10	1048	00:00:00.04
p7	942.57	8	1458	00:12:48.22
p8	2415.86	24	3222	00:25:47.94
p9	975.16	8	1117	00:00:00.06
p10	1866.63	17	2477	00:17:14.28
p11	962.57	17	2562	00:19:44.72
p12	1813.69	13	804	00:18:04.66
p13	4668.41	61	104585	01:18:23.77
p14	1038.58	8	90	00:00:02.86
p15	2121.3	8	179	00:01:39.99
p16	3052.67	8	222	00:03:58.32
p17	1790.77	14	188	00:03:32.64
p18	3407.95	16	338	00:14:57.85
p19	5166.64	16	496	00:27:51.35
p20	9234.58	16	778	00:56:01.73
p21	2711.9	20	298	00:07:17.27
p22	5124.47	23	434	00:32:43.93
p23	7682.59	23	568	00:57:13.55
p24	4090.13	16	254	00:00:57.74
p25	4883.9	17	205	00:00:39.87
p26	4831.98	17	236	00:00:34.00
p27	28171.66	32	406	00:12:39.97
p28	25970.76	34	529	00:13:10.65
p29	26225.95	35	536	00:14:08.96
p30	101251.85	45	613	00:35:00.10
p31	103225.77	48	681	00:35:30.71
p32	101959.86	50	737	00:35:05.65
pr01	2287.31	8	1229	00:02:11.54
pr02	4292.8	15	1402	00:22:45.85
pr03	5971.25	23	2968	00:47:03.42
pr04	7789.51	28	3991	09:36:58.07
pr05	7942.64	40	4876	01:32:37.10
pr06	9693.21	43	5339	01:52:56.01
pr07	5190.04	18	2590	00:19:50.11
pr08	7756.54	36	5493	01:24:06.61
pr09	12251.13	49	6390	02:03:30.73
pr10	17485.14	72	8429	03:11:13.44
			Sum in h	30.29

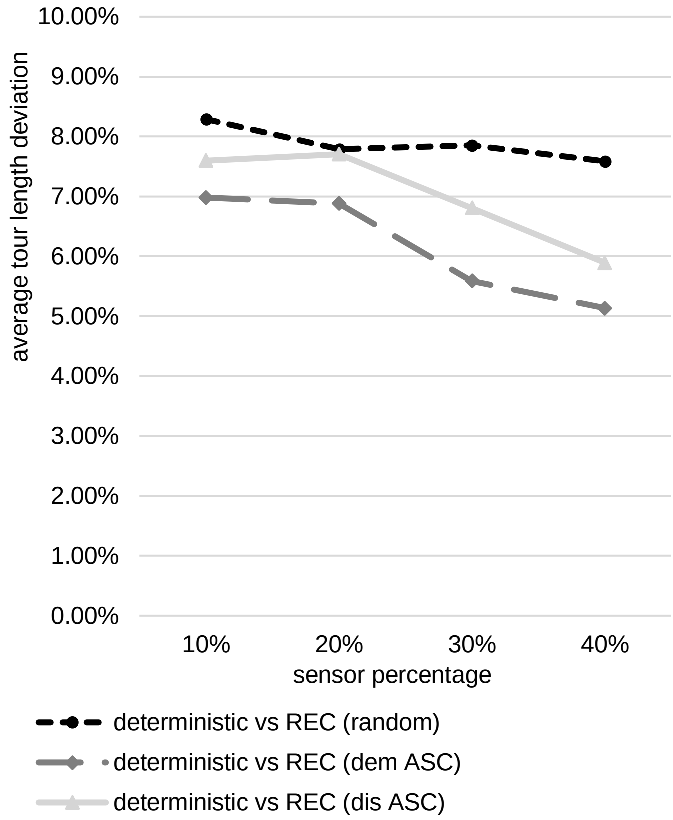
Note: reoptimization after every customer has been used in combination with random sensor assignment

In Figure 25 the average tour length deviations between the setting with deterministic and stochastic demands for differing sensor levels and different forms of sensor assignment for the REC approach are displayed. The results for every instance of REC are summarized in Appendix M. For the distance-based and demand-based sensor assignment, only the waste bins with the furthest distance from the depot and the highest demand respectively receive the sensors, as the previous experiment with reoptimization after delivery failure has shown, that the reverse ordering of sensors performed the worst and good results cannot be expected for assigning sensors to waste bins with the shortest distance to the depot or lowest demand, respectively.

The results for the random sensor assignment are ambiguous, as there is an increase in the average tour length deviation from 30% to 40% sensors as would be expected, but then the deviation decreases with a lower sensor level and increases again for the lowest sensor setting. In addition, the performance is worse than almost all other sensor allocation paradigms, except for the demand-based sensor assignment, where customers with low demands receive sensors. However, the demand-based and distance-based sensor assignment outperform the random sensor allocation with REC for every sensor level. It is especially noteworthy, that the demand-based sensor assignment with REC dominates random and distance-based sensor assignment at every level. Additionally, the demand-based sensor allocation with REC and 30% and 40% sensor had the lowest average tour length deviation of all sensor assignment methods for its respective sensor level.

Figure 25

Average tour length deviation deterministic vs REC



Note: the average tour length deviations are computed between all instances with deterministic and stochastic demands with varying sensor levels
Reoptimization after every customer is used with random sensor assignment as well as demand- and distance-based sensor allocation

5.4.5 Summary of the Results

All of the results for the average tour length deviation in the setting with low sensor levels and for both recourse policies, DTD and reoptimization approaches are presented in Table 12. It can be seen that the methods distance-based and demand-based assignment performed better than the random assignment, when bins are assigned with sensors that are furthest from the depot or have the highest demand, respectively. The distance-based sensor allocation (ASC) performed best for 10% sensor allocation, while the demand-based sensor allocation (ASC) beat all other forms of assignment for 20% of sensors and up. This holds true for the DTD recourse policy, as well as the RDF.

However, assigning the nearest bins or bins with lowest demand with sensors had the adverse effect and increased the average tour lengths more than with the random sensor allocation. The RDF approach allowed to decrease the average tour length deviation in every case. At the same time, the REC approach achieved worse results for the random sensor assignment, however the demand-based sensor allocation with REC outperformed all other methods for 30% and 40% of sensors.

Table 12

Comparison of average tour lengths deviations for each sensor level and the overall best results

Form of Assignment	**Sensor Level**			
	10%	20%	30%	40%
Recourse Policy: Detour to Depot (DTD)				
random sensor assignment	X**	7.51%	X**	6.50%
Demand-based sensor assignment (ASC)	**6.53%**	**6.28%**	**5.98%**	5.72%
Demand-based sensor assignment (DESC)	8.11%	8.35%	8.47%	8.00%
Distance-based sensor assignment (ASC)	7.85%	7.31%	6.66%	**5.55%**
Distance-based sensor assignment (DESC)	8.63%	7.70%	7.17%	6.78%
Recourse Policy: Reoptimization after Delivery Failure (RDF)				
random sensor assignment	X**	7.23%	X**	6.45%
Demand-based sensor assignment (ASC)	**6.37%**	**6.21%**	**5.84%**	5.44%
Demand-based sensor assignment (DESC)	7.99%	8.08%	8.23%	7.69%
Distance-based sensor assignment (ASC)	7.70%	7.16%	6.38%	**5.30%**
Distance-based sensor assignment (DESC)	8.30%	7.57%	6.97%	6.59%
Recourse Policy: Reoptimization after Each Customer (REC)				
random sensor assignment	8.29%	7.79%	7.85%	7.59%

Demand-based sensor assignment (ASC)	**6.98%**	**6.88%**	**5.59%**	**5.14%**
Distance-based sensor assignment (ASC)	7.60%	7.70%	6.81%	5.90%

Note: the average tour length deviation is based on the tour lengths in the original dataset with deterministic demands and the new datasets with partial stochastic demands depending on the sensor level.
**results are not presented, as the random sensor assignment has only been done for 20-40-60-80% sensors

5.5 Managerial Insights

The findings of this thesis can be beneficial for practitioners in logistics and supply chain management, who are working in the field of smart cities and/or SWM and have important managerial implications. When a project with the aim of improving the routing in SWM is conducted, equipping waste bins with sensors should be considered, as this can reduce the overall tour lengths in solid waste collection. Moreover, if the budget is limited and only a small number of sensors can be placed, practitioners should be careful with the sensor placement and should allocate them to waste bins with high demands, if waste levels are highly variable. Provided that this is not the case, tour length improvements can still be achieved, especially when sensors are equipped to the bins furthest from the depot. At the same time, sensors should not be allocated to bins furthest away from the depot or to bins with the lowest demands, as the results are worse than when the sensors are allocated randomly.

Further improvements can also be attained, when a reoptimization procedure is done after each delivery failure. This presupposes that vehicles can communicate with each other and that the routes can be flexibly adapted. However, practitioners need to be careful with route plannings that are adapted after each visited customer. A lot of computational power is needed to calculate potential improvements after each step. However, it could be seen that this can prove worthwhile for the distance-based and demand-based sensor assignment, especially for higher levels of smart waste bins. Overall, these findings emphasize the importance of strategic allocation of sensors to waste bins in IoT-based solid waste management to minimize the overall tour lengths and improve the operational efficiency.

6 Limitations

The generated results with the Tabu Search method did not produce better results than in previous algorithms. This might be due to the memory consuming nature of saving whole solutions of neighbors, which impacted the efficiency of the algorithm. Secondly, the absence of more sophisticated but time-consuming local search algorithms also inhibited the generation of better results. However, all instances could be solved with the algorithm and the results only served as starting point for the analysis of the sensor placement, which was the main focus of this study.

Moreover, it should be noted that the demand variability of customers in SWM usually is not that severe, therefore demand-based ordering might not be as impactful as presented in this thesis. However, using instances with identical demands for each customer as basis for assigning stochastic demands might have affected the comparability to the original instances.

Another aspect that has not been considered in this thesis is the lack of a dumping site for emptying the waste bins, as the instances used were not designed with this intention.

Finally, the results are all based on the same seed for assigning demands of the stochastic customers, making them easily comparable across the different forms of sensor assignment. Nonetheless, only one seed was used for each instance, which could be problematic, if the demands assigned are particularly improbable.

Overall, these limitations suggest that further research is needed to improve the efficiency of the algorithm used for solving the PVRP instances. Furthermore, for solving the sensor allocation problem in IoT-based SWM, complexities of the real word have to be considered to a greater extent for example the results should be checked for several demand distributions.

7 Conclusion

This research has explored the Periodic Vehicle Routing Problem in solid waste management and aimed at finding suitable ways to allocate smart IoT-based sensors to waste bins in order to decrease the overall tour lengths and increase the operational efficiency.

To solve this problem, at the first a Tabu Search algorithm has been constructed to solve PVRP instances from the literature, which served as the basis for the comparison of PVRP solutions with stochastic demands. Subsequently, the PVRP instances have been modified and waste bins with stochastic demands have been introduced. In the given instances, waste bins have been assigned with various levels of sensors, where demands are deterministic. To show the influence of the different forms of sensor allocations on the total tour lengths, a random sensor allocation has been used as a baseline. Furthermore, a demand-based, as well as distance-based sensor allocation were employed for the comparison with the random sensor allocation. Finally, three forms of recourse policies have been utilized to show their impact on the overall tour length.

The results showed that the random assignment of sensors performed the worst, while the demand-based sensor assignment performed best. However, the waste bins' demand variability in the experimental setting was more pronounced than usually in solid waste management, due to the varying demands in the problem sets used. Therefore, it might be more profitable to use a distance-based sensor assignment when the demand variability is low. However, further research would be needed to verify this assumption.

Regarding the recourse policies, reoptimization after delivery failure allowed to decrease the tour lengths significantly compared to the detour-to-depot recourse policy, but the reoptimization after every customer performed even better for sensor levels of 30-40%. Nonetheless, practitioners should be careful in selecting their method of reoptimization, as reoptimization after every customer can be computationally expensive and takes a long time to deliver satisfactory results.

In conclusion, this thesis has contributed to the body of research, because the limited allocation of sensors to waste bins has not yet been extensively discussed in the literature. By comparing the effect of various sensor allocation methods coupled with different recourse policies, valuable insights for the vehicle routing in solid waste management could be generated, that can be used to improve the operational efficiency.

8 References

Ak, A., & Erera, A. L. (2007). A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transportation Science*, *41*(2), 222–237. https://doi.org/10.1287/trsc.1060.0180

Akram, S. V., Singh, R., AlZain, M. A., Gehlot, A., Rashid, M., Faragallah, O. S., El-Shafai, W., & Prashar, D. (2021). Performance analysis of IoT and long-range radio-based sensor node and gateway architecture for solid waste management. *Sensors*, *21*(8), 1–22. https://doi.org/10.3390/s21082774

Anagnostopoulos, T., Zaslavsky, A., Kolomvatsos, K., Medvedev, A., Amirian, P., Morley, J., & Hadjieftymiades, S. (2017). Challenges and opportunities of waste management in IoT-enabled smart cities: A survey. *IEEE Transactions on Sustainable Computing*, *2*(3), 275–289. https://doi.org/10.1109/TSUSC.2017.2691049

Angelelli, E., & Grazia Speranza, M. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research*, *137*(2), 233–247. https://doi.org/10.1016/S0377-2217(01)00206-5

Baldo, D., Mecocci, A., Parrino, S., Peruzzi, G., & Pozzebon, A. (2021). A multi-layer LoRaWAN infrastructure for smart waste management. *Sensors*, *21*(8), 1–27. https://doi.org/10.3390/s21082600

Borghetti, A., Antonio, F., Lacalandra, F., Lodi, A., Martello, S., Nucci, C. A., & Trebbi, A. (2001). Lagrangian relaxation and Tabu Search approaches for the unit commitment problem. In *IEEE Porto Power Tech Proceedings* (Vol. 3, 7pp.). https://doi.org/10.1109/PTC.2001.964914

Bräysy, O., & Gendreau, M. (2002). Tabu Search heuristics for the vehicle routing problem with time windows. *Top*, *10*(2), 211–237. https://doi.org/10.1007/BF02579017

Campbell, A. M., & Wilson, J. H. (2014). Forty years of periodic vehicle routing. *Networks*, *63*(1), 2–15. https://doi.org/10.1002/net.21527

Cerchecci, M., Luti, F., Mecocci, A., Parrino, S., Peruzzi, G., & Pozzebon, A. (2018). A low power IoT sensor node architecture for waste management within smart cities context. *Sensors*, *18*(4), 1–27. https://doi.org/10.3390/s18041282

Cerrone, C., & Sciomachen, A. (2022). Vrp in urban areas to optimize costs while mitigating environmental impact. *Soft Computing*, *26*(19), 10223–10237. https://doi.org/10.1007/s00500-022-07325-z

Chao, I.-M., Golden, B. L., & Wasil, E. (1995). An improved heuristic for the period vehicle routing problem. *Networks, 26*(1), 25–44. https://doi.org/10.1002/net.3230260104

Chiew, Y. L., & Brunklaus, B. (2021). *Life cycle assessment of Internet of Things (IoT) solution in Södertälje municipality: A smart waste collection system.* RISE Research Institutes of Sweden. http://ri.diva-portal.org/smash/get/diva2:1601326/FULLTEXT01.pdf

Christiansen, C. H., Lysgaard, J., & Wøhlk, S. (2009). A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands. *Operations Research Letters, 37*(6), 392–398. https://doi.org/10.1016/j.orl.2009.05.008

Clarke, G., & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research, 12*(4), 568–581. https://doi.org/167703

Cook, T. M., & Russell, R. A. (1978). A simulation and statistical analysis of stochastic vehicle routing with timing constraints. *Decision Sciences, 9*(4), 673–687. https://doi.org/10.1111/j.1540-5915.1978.tb00753.x

Cordeau, J.-F., Gendreau, M., & Laporte, G. (1997). A Tabu Search heuristic for periodic and multi-depot vehicle routing problems. *Networks*, *30*(2). https://doi.org/10.1002/(SICI)1097-0037(199709)30:2<105::AID-NET5>3.0.CO;2-G

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

Dayarian, I., Crainic, T. G., Gendreau, M., & Rei, W. (2016). An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review, 95*, 95–123. https://doi.org/10.1016/j.tre.2016.09.004

Doan, T. T., Bostel, N., & Hà, M. H. (2021). The vehicle routing problem with relaxed priority rules. *EURO Journal on Transportation and Logistics, 10*(100039), 1–15. https://doi.org/10.1016/j.ejtl.2021.100039

Eilon, S., C. D. T. Watson-Gandy, & Christofides, N. (1974). Distribution management-mathematical modelling and practical analysis. *IEEE Transactions on Systems, Man, and Cybernetics*(SMC-4), Article 6, 589. https://doi.org/10.1109/TSMC.1974.4309370

Esmaeilian, B., Wang, B., Lewis, K., Duarte, F., Ratti, C., & Behdad, S. (2018). The future of waste management in smart and sustainable cities: A review and concept paper. *Waste Management*, *81*, 177–195. https://doi.org/10.1016/j.wasman.2018.09.047

Florio, A. M., Feillet, D., Poggi, M., & Vidal, T. (2022). Vehicle routing with stochastic demands and partial reoptimization. *Transportation Science*, *56*(5), 1393–1408. https://doi.org/10.1287/trsc.2022.1129

Florio, A. M., Hartl, R. F., & Minner, S. (2020). Optimal a priori tour and restocking policy for the single-vehicle routing problem with stochastic demands. *European Journal of Operational Research*, *285*(1), 172–182. https://doi.org/10.1016/j.ejor.2018.10.045

Gauvin, C., Desaulniers, G., & Gendreau, M. (2014). A branch-cut-and-price algorithm for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, *50*, 141–153. https://doi.org/10.1016/j.cor.2014.03.028

Gendreau, M., Hertz, A., & Laporte, G. (1994). A Tabu Search heuristic for the vehicle routing problem. *Management Science*(40), Article 10, 1276–1290. https://www.jstor.org/stable/2661622

Gendreau, M., Jabali, O., & Rei, W. (2016). 50th Anniversary invited article—Future research directions in stochastic vehicle routing. *Transportation Science*, *50*(4), 1163–1173. https://doi.org/10.1287/trsc.2016.0709

Gendreau, M., Laporte, G., & Seguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research*(88), 3–12. https://doi.org/10.5194/acp-2016-545-RC1

Gendreau, M., & Potvin, J.-Y. (2010). *Handbook of Metaheuristics* (Vol. 146). Springer US. https://doi.org/10.1007/978-1-4419-1665-5

Glover, F. (1989). Tabu Search - Part I. *ORSA Journal on Computing*(1), Article 3, 190–206. https://doi.org/10.1287/ijoc.1.3.190

Glover, F., & Laguna, M. (1998). Tabu Search. *Handbook of Combinatorial Optimization*, 2093–2229. https://doi.org/10.1007/978-1-4613-0303-9_33

Glover, F., & Taillard, E. (1993). A user's guide to tabu search. *Annals of Operations Research*, *41*(1), 1–26. https://doi.org/10.1007/BF02078647

Golden, B. L., Raghavan, S., Wasil, E. A., & others. (2008). *The vehicle routing problem: Latest advances and new challenges* (Vol. 43). Springer. https://doi.org/10.1007/978-0-387-77778-8

Goodson, J. C., Ohlmann, J. W., & Thomas, B. W. (2012). Cyclic-order neighborhoods with application to the vehicle routing problem with stochastic demand. *European Journal of Operational Research*, *217*(2), 312–323. https://doi.org/10.1016/j.ejor.2011.09.023

Gutin, G., Yeo, A., & Zverovich, A. (2001). Traveling salesman should not be greedy: Domination analysis of greedy-type heuristics for the TSP. *Discrete Applied Mathematics*, *117*(1-3), 81–86. https://doi.org/10.1016/S0166-218X(01)00195-0.

HEC Montreal. (n.d.). *Data File Structure README.txt*. Retrieved January 21, 2023, from https://chairelogistique.hec.ca/data/README.TXT

Hemmelmayr, V. C., Doerner, K. F., & Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research*, *195*(3), 791–802. https://doi.org/10.1016/j.ejor.2007.08.048

Hemmelmayr, V. C., Doerner, K. F., Hartl, R. F., & Vigo, D. (2014). Models and algorithms for the integrated planning of bin allocation and vehicle routing in solid waste management. *Transportation Science*, *48*(1), 103–120. https://doi.org/10.1287/trsc.2013.0459

Ho, S. C., & Haugland, D. (2004). A Tabu Search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, *31*(12), 1–18. https://doi.org/10.1016/S0305-0548(03)00155-2

Johansson, O. M. (2006). The effect of dynamic scheduling and routing in a solid waste management system. *Waste Management*, *26*(8), 875–885. https://doi.org/10.1016/j.wasman.2005.09.004

Johansson, O. M., & Johansson, R. (2008). Model predictive control for scheduling and routing in a solid waste management system. *IFAC Proceedings Volumes*, *41*(2), 4481–4486. https://doi.org/10.3182/20080706-5-KR-1001.00755

Karthik, M., Sreevidya, L., Nithya Devi, R., Thangaraj, M., Hemalatha, G., & Yamini, R. (2021). An efficient waste management technique with IoT based smart garbage system. *Materials Today: Proceedings, 2021*. https://doi.org/10.1016/j.matpr.2021.07.179

Keeler, H. P. (2019). *Simulating Poisson random variables – Direct method*. https://hpaulkeeler.com/simulating-poisson-random-variables-direct-method/

Kim, G., Ong, Y.-S., Heng, C. K., Tan, P. S., & Zhang, N. A. (2015). City vehicle routing problem (City VRP): A review. *IEEE Transactions on Intelligent Transportation Systems*, *16*(4), 1654–1666. https://doi.org/10.1109/TITS.2015.2395536

Konstantakopoulos, G. D., Gayialis, S. P., & Kechagias, E. P. (2022). Vehicle routing problem and related algorithms for logistics distribution: A literature review and classification. *Oper Res Int J*, *22*, 2033–2062. https://doi.org/10.1007/s12351-020-00600-7

Laporte, G. (1992). The vehicle routing problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, *59*(3), 345–358. https://doi.org/10.1016/0377-2217(92)90192-C

Laporte, G., Louveaux, F. V., & van Hamme, L. (2002). An integer L-shaped algorithm for the capacitated vehicle routing problem with stochastic demands. *Operations Research*, *50*(3), 415–423. https://doi.org/10.1287/opre.50.3.415.7751

Laporte, G., Ropke, S., & Vidal, T. (2014). Chapter 4: Heuristics for the vehicle routing problem. In P. Toth & D. Vigo (Eds.), *MOS-SIAM series on optimization: Vol. 18. Vehicle routing: Problems, methods, and applications* (pp. 87–116). SIAM Society for Industrial and Applied Mathematics; Mathematical Optimization Society. https://doi.org/10.1137/1.9781611973594.ch4

Lopes, M., & Ramos, T. R. (2023). Efficient sensor placement and online scheduling of bin collection. *Computers & Operations Research*, *151*(106113). https://doi.org/10.1016/j.cor.2022.106113

Lysgaard, J. (1997). Clarke & Wright's savings algorithm. *Department of Management Science and Logistics, the Aarhus School of Business*(44), 1–7. https://www.academia.edu/27827242/Clarke_and_Wrights_Savings_Algorithm

Mańdziuk, J., & Nejman, C. (2015). UCT-based approach to capacitated vehicle routing problem. In L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. A. Zadeh, & J. M. Zurada (Eds.), *Artificial Intelligence and Soft Computing* (pp. 679–690). Springer International Publishing.

Mishra, A., & Ray, A. K. (2020). IoT cloud-based cyber-physical system for efficient solid waste management in smart cities: A novel cost function based route optimisation technique for waste collection vehicles using dustbin sensors and real-time road traffic informatics. *IET Cyber-Physical Systems: Theory & Applications*, *5*(4), 330–341. https://doi.org/10.1049/iet-cps.2019.0110

Omara, A., Gulen, D., Kantarci, B., & Oktug, S. (2018). Trajectory assisted municipal agent mobility: A sensor-driven smart waste management system. *Journal of Sensor and Actuator Networks*, *7*(3), 1–29. https://doi.org/10.3390/jsan7030029

Oyola, J., Arntzen, H., & Woodruff, D. L. (2018). The stochastic vehicle routing problem, a literature review, part I: Models. *EURO Journal on Transportation and Logistics*, *7*(3), 193–221. https://doi.org/10.1007/s13676-016-0100-5

Ramos, T. R. P., Morais, C. S. de, & Barbosa-Póvoa, A. P. (2018). The smart waste collection routing problem: Alternative operational management approaches. *Expert Systems with Applications*, *103*, 146–158. https://doi.org/10.1016/j.eswa.2018.03.001

Rocki, K., & Suda, R. (2012). Accelerating 2-opt and 3-opt local search using GPU in the travelling salesman problem. In *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)* (pp. 705–706). IEEE. https://doi.org/10.1109/CCGrid.2012.133

Roy, A., Manna, A., Kim, J., & Moon, I. (2022). IoT-based smart bin allocation and vehicle routing in solid waste management: A case study in South Korea. *Computers & Industrial Engineering*, *171*(108457), 1–17. https://doi.org/10.1016/j.cie.2022.108457

Russell, R. A., & Gribbin, D. (1991). A multiphase approach to the period routing problem. *Networks*, *21*(7), 747–765. https://doi.org/10.1002/net.3230210704

Russell, R. A., & Igo, W. (1979). An assignment routing problem. *Networks*, *9*(1), 1–17. https://doi.org/10.1002/net.3230090102

Schedl, M., & Strauss, C. (2011). A periodic routing problem with stochastic demands. In *2011 International Conference on Complex, Intelligent, and Software Intensive Systems* (pp. 350–357). IEEE. https://doi.org/10.1109/CISIS.2011.57

Secomandi, N., & Margot, F. (2009). Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, *57*(1), 214–230. https://doi.org/10.1287/opre.1080.0520

Silver, E. A. (2004). An overview of heuristic solution methods. *Journal of the Operational Research Society*, *55*(9), 936–956. https://doi.org/10.1057/palgrave.jors.2601758

Sohag, M. U., & Podder, A. K. (2020). Smart garbage management system for a sustainable urban life: An IoT based application. *Internet of Things*, *11*(100255), 1–15. https://doi.org/10.1016/j.iot.2020.100255

Soriano, P., & Gendreau, M. (1996). Diversification strategies in Tabu Search algorithms for the maximum clique problem. *Annals of Operations Research*, *63*(2), 189–207. https://doi.org/10.1007/BF02125454

Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, *23*(8), 661–673. https://doi.org/10.1002/net.3230230804

Tan, S.-Y., & Yeh, W.-C. (2021). The vehicle routing problem: State-of-the-art classification and review. *Applied Sciences, 11*(21), 10295. https://doi.org/10.3390/app112110295

Tavares, L. G., Lopes, H. S., & Lima, C. R. E. (2009). Construction and improvement heuristics applied to the capacitated vehicle routing problem. In *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)* (pp. 690–695). IEEE. https://doi.org/10.1109/NABIC.2009.5393467

Toth, P., & Vigo, D. (Eds.). (2014). *MOS-SIAM series on optimization: Vol. 18. Vehicle routing: Problems, methods, and applications* (Second edition). SIAM Society for Industrial and Applied Mathematics; Mathematical Optimization Society. https://doi.org/10.1137/1.9781611973594

Yang, W.-H., Mathur, K., & Ballou, R. H. (2000). Stochastic vehicle routing problem with restocking. *Transportation Science*, *34*(1), 99–112. https://doi.org/10.1287/trsc.34.1.99.12278

Yee, J. R., & Golden, B. L. (1980). A note on determining operating strategies for probabilistic vehicle routing. *Naval Research Logistics Quarterly*, *27*(1), 159–163. https://doi.org/10.1002/nav.3800270114

Zäpfel, G., Braune, R., & Bögl, M. (2010). *Metaheuristic search concepts: A tutorial with applications to production and logistics*. Springer. https://doi.org/10.1007/978-3-642-11343-7

9 Appendix

Appendix A

Data File structure, part 1 (HEC Montreal)

```
The format of data and solution files in all directories (except "darp", "gmstp", "gqap") is as follows:

A) DATA FILES

The first line contains the following information:

        type m n t

where

        type = 0 (VRP)
               1 (PVRP)
               2 (MDVRP)
               3 (SDVRP)
               4 (VRPTW)
               5 (PVRPTW)
               6 (MDVRPTW)
               7 (SDVRPTW)

        m = number of vehicles

        n = number of customers

        t = number of days (PVRP), depots (MDVRP) or vehicle types (SDVRP)

The next t lines contain, for each day (or depot or vehicle type), the
following information:

        D Q

where

        D = maximum duration of a route

        Q = maximum load of a vehicle
```

Appendix B

Data File structure, part 2 (HEC Montreal)

```
The next lines contain, for each customer, the following information:

        i x y d q f a list e l

where

        i = customer number

        x = x coordinate

        y = y coordinate

        d = service duration

        q = demand

        f = frequency of visit

        a = number of possible visit combinations

        list = list of all possible visit combinations

        e = beginning of time window (earliest time for start of service),
            if any

        l = end of time window (latest time for start of service), if any

               Each visit combination is coded with the decimal equivalent of
               the corresponding binary bit string. For example, in a 5-day
               period, the code 10 which is equivalent to the bit string 01010
               means that a customer is visited on days 2 and 4. (Days are
               numbered from left to right.)

Note : In the case of the MDVRP, the lines go from 1 to n + t and the last t
entries correspond to the t depots. In the case of the VRP, PVRP and SDVRP, the
lines go from 0 to n and the first entry corresponds to the unique depot.


B) SOLUTION FILES

The first line contains the cost of the solution (total duration excluding
service time).

The next lines contain, for each route, the following information:

        l k d q list

where

        l = number of the day (or depot or vehicle type)

        k = number of the vehicle

        d = duration of the route

        q = load of the vehicle

        list = ordered sequence of customers (with start-of-service times, if
               applicable)
```

Appendix C

tour length comparison deterministic vs random sensor allocation (20-100% stochasticity)

dataset	deterministic	stoch 20	stoch 40	stoch 60	stoch 80	stoch 100	Reopt stoch 20	Reopt stoch 40	Reopt stoch 60	Reopt stoch 80	Reopt stoch 100
p1	597.54	627.06	643.71	667.79	597.54	726.29	627.06	643.71	597.54	597.54	726.29
p2	1438.38	1515.67	1531.78	1438.38	1544.3	1504.51	1515.67	1531.78	1544.18	1544.3	1488.48
p3	628.61	648.61	628.61	648.61	692.42	628.61	648.61	628.61	628.61	692.42	628.61
p4	945.57	1083.93	1007.36	1127.48	980.69	1053.69	1083.93	1007.36	966.13	959.31	1053.69
p5	2177.77	2330.04	2467.74	2310.22	2406.95	2334.35	2330.04	2465.86	2476.62	2406.95	2324.89
p6	972.89	1059.54	1105.99	1042.07	1040.83	1071.19	1059.54	1105.99	1034.67	1040.83	1071.19
p7	942.57	952.57	964.93	942.57	942.57	942.57	952.32	964.93	952.57	942.57	942.57
p8	2263.63	2287.71	2401.25	2442.36	2415.91	2410.44	2287.71	2423.75	2445.06	2415.91	2378.62
p9	916.74	977.04	958.93	916.74	952.8	958.93	977.04	958.93	916.74	952.8	958.93
p10	1762.76	1827.69	1860.72	1846.33	1860.72	1990.12	1827.69	1860.72	1887.62	1860.72	1990.12
p11	864.80	935.41	971.23	969.47	966.37	1038.64	935.41	971.23	916.15	966.37	1038.64
p12	1321.68	1623.74	1483.96	1624.16	1620.94	1638.55	1572.31	1480.17	1598.96	1579.73	1629.7
p13	4352.19	4545.54	4493.8	4611.56	4688.35	4765.4	4534.71	4484.50	4583.84	4634.98	4762.18
p14	958.58	958.58	1018.58	1018.58	978.58	978.58	958.58	1018.58	998.58	978.58	978.58
p15	1981.30	2081.3	2061.3	2101.3	2101.31	2161.31	2080.89	2061.30	2061.3	2099.44	2161.31
p16	2972.67	2992.67	2992.67	3032.68	3072.67	3112.68	2992.67	2992.67	2992.67	3072.67	3112.68
p17	1630.77	1670.77	1750.77	1670.77	1770.77	1790.77	1670.77	1744.68	1788.45	1770.77	1790.77
p18	3267.95	3287.95	3287.95	3287.95	3367.95	3287.95	3287.95	3287.95	3447.95	3366.5	3287.95
p19	5046.64	5046.64	5086.64	5186.64	5086.64	5206.64	5046.64	5086.64	5286.64	5086.64	5206.64
p20	9094.58	9274.58	9274.58	9554.58	9114.58	9734.58	9274.58	9274.58	9354.58	9114.58	9732.98
p21	2291.90	2411.9	2471.9	2511.9	2571.9	2551.9	2409.42	2471.90	2791.9	2541.59	2519.71
p22	4524.47	4784.47	4944.48	4844.47	5244.47	4944.47	4779.35	4941.80	4906.4	5178.94	4943.64
p23	6962.60	7082.6	7502.6	7062.6	7462.6	6962.6	7082.6	7492.33	7322.61	7462.6	6962.6
p24	3859.19	4144.37	4500.48	4285.6	4170.13	4250.13	4144.37	4500.48	4144.36	4170.13	4250.13
p25	3907.61	4138.55	4809.89	4539.19	4773.6	4568.43	4138.55	4809.89	4468.96	4773.6	4501.24
p26	3974.63	4594.75	4450.65	4170.1	4792.53	4677.06	4563.68	4450.65	4405.57	4792.53	4675.76
p27	23990.77	25843.59	27107.74	30389.78	26417.76	28969.35	25843.59	27060.76	27507.57	26206.44	28962.38
p28	24260.25	25037.6	26236.83	26223.24	27628.21	27200.59	24857.09	26169.64	27376.55	27537.39	27092.42
p29	24971.93	25864.75	27073.04	26442.1	28524.73	28551.5	25864.75	26995.69	28305.74	28512.76	28551.5
p30	85909.78	91626.34	100217.1	96335.21	101578.24	101230.26	91626.34	100042.72	96374.49	101578.24	101139.97
p31	84525.69	90767.14	100719.72	94690.32	99957.82	92377.39	90767.14	99638.19	101452.88	99957.82	92352.47
p32	87642.98	94786.76	98171.76	101585.51	105868.89	102297.79	94057.69	97818.59	96435.21	104649.71	101011.37
pr01	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31
pr02	4146.13	4209.35	4199.61	4250.52	4279.1	4248.02	4201.18	4199.61	4221.4	4278.66	4240.94
pr03	5598.49	5642.99	5733.38	5874.78	5925.95	5896.56	5642.99	5733.38	5831.77	5925.95	5895.93
pr04	7360.78	7449.82	7750.56	7546.32	7827.28	7663.18	7449.82	7750.56	7790.99	7827.28	7663.18
pr05	7362.97	7741.38	7997.22	7648.07	7822.29	8014.34	7741.38	7930.89	7744.8	7783.37	7969.92
pr06	8817.96	9523.36	9986.67	9714.85	9718.44	9816.7	9523.36	9986.67	9595.68	9718.44	9816.7
pr07	5108.60	5190.04	5124.43	5108.6	5190.04	5153.21	5190.04	5124.43	5160.35	5190.04	5153.21
pr08	7581.11	7757.52	7873.78	7972.45	7756.54	7963.17	7744.83	7837.01	7746.98	7730.38	7944.56
pr09	11061.23	11732.52	11922.19	12246.87	12047.34	12378.14	11725.88	11853.88	11691.69	11997.61	12196.91
pr10	15913.03	17137.32	17023.52	17262.12	17250.55	17451.79	16987.62	16970.70	16830.69	17095.69	17419.37
Avg	**11338.02**	**12035.27**	**12716.60**	**12605.53**	**12983.30**	**12780.71**	**12006.98**	**12668.12**	**12639.83**	**12935.24**	**12733.72**
σ^2	**466432863.76**	**538027183.24**	**626798141.69**	**603267843.16**	**663269146.23**	**617743676.63**	**535068699.91**	**620021004.40**	**609235468.59**	**657787862.71**	**611835995.09**
σ	**21597.06**	**23195.41**	**25035.94**	**24561.51**	**25754.01**	**24854.45**	**23131.55**	**24900.22**	**24682.70**	**25647.38**	**24735.32**

Note: results on the left (stoch 20…100 = DTD); results on the right (Reopt stoch 20…100 = RDF)

Appendix D

random assignment tour length % deviation stochastic vs deterministic demands (20-100% stochasticity)

dataset	det vs stoch 20	det vs stoch 40	det vs stoch 60	det vs stoch 80	det vs stoch 100	det vs reopt 20	det vs reopt 40	det vs reopt 60	det vs reopt 80	det vs reopt 100
p1	4.71%	7.17%	10.52%	0.00%	17.73%	4.71%	7.17%	0.00%	0.00%	17.73%
p2	5.10%	6.10%	0.00%	6.86%	4.40%	5.10%	6.10%	6.85%	6.86%	3.37%
p3	3.08%	0.00%	3.08%	9.22%	0.00%	3.08%	0.00%	0.00%	9.22%	0.00%
p4	12.76%	6.13%	16.13%	3.58%	10.26%	12.76%	6.13%	2.13%	1.43%	10.26%
p5	6.54%	11.75%	5.73%	9.52%	6.71%	6.54%	11.68%	12.07%	9.52%	6.33%
p6	8.18%	12.03%	6.64%	6.53%	9.18%	8.18%	12.03%	5.97%	6.53%	9.18%
p7	1.05%	2.32%	0.00%	0.00%	0.00%	1.02%	2.32%	1.05%	0.00%	0.00%
p8	1.05%	5.73%	7.32%	6.30%	6.09%	1.05%	6.61%	7.42%	6.30%	4.83%
p9	6.17%	4.40%	0.00%	3.78%	4.40%	6.17%	4.40%	0.00%	3.78%	4.40%
p10	3.55%	5.26%	4.53%	5.26%	11.42%	3.55%	5.26%	6.61%	5.26%	11.42%
p11	7.55%	10.96%	10.80%	10.51%	16.74%	7.55%	10.96%	5.60%	10.51%	16.74%
p12	18.60%	10.94%	18.62%	18.46%	19.34%	15.94%	10.71%	17.34%	16.34%	18.90%
p13	4.25%	3.15%	5.62%	7.17%	8.67%	4.02%	2.95%	5.05%	6.10%	8.61%
p14	0.00%	5.89%	5.89%	2.04%	2.04%	0.00%	5.89%	4.01%	2.04%	2.04%
p15	4.80%	3.88%	5.71%	5.71%	8.33%	4.79%	3.88%	3.88%	5.63%	8.33%
p16	0.67%	0.67%	1.98%	3.25%	4.50%	0.67%	0.67%	0.67%	3.25%	4.50%
p17	2.39%	6.85%	2.39%	7.91%	8.93%	2.39%	6.53%	8.82%	7.91%	8.93%
p18	0.61%	0.61%	0.61%	2.97%	0.61%	0.61%	0.61%	5.22%	2.93%	0.61%
p19	0.00%	0.79%	2.70%	0.79%	3.07%	0.00%	0.79%	4.54%	0.79%	3.07%
p20	1.94%	1.94%	4.81%	0.22%	6.57%	1.94%	1.94%	2.78%	0.22%	6.56%
p21	4.98%	7.28%	8.76%	10.89%	10.19%	4.88%	7.28%	17.91%	9.82%	9.04%
p22	5.43%	8.49%	6.61%	13.73%	8.49%	5.33%	8.44%	7.78%	12.64%	8.48%
p23	1.69%	7.20%	1.42%	6.70%	0.00%	1.69%	7.07%	4.92%	6.70%	0.00%
p24	6.88%	14.25%	9.95%	7.46%	9.20%	6.88%	14.25%	6.88%	7.46%	9.20%
p25	5.58%	18.76%	13.91%	18.14%	14.46%	5.58%	18.76%	12.56%	18.14%	13.19%
p26	13.50%	10.70%	4.69%	17.07%	15.02%	12.91%	10.70%	9.78%	17.07%	14.99%
p27	7.17%	11.50%	21.06%	9.19%	17.19%	7.17%	11.34%	12.78%	8.45%	17.17%
p28	3.10%	7.53%	7.49%	12.19%	10.81%	2.40%	7.30%	11.38%	11.90%	10.45%
p29	3.45%	7.76%	5.56%	12.46%	12.54%	3.45%	7.50%	11.78%	12.42%	12.54%
p30	6.24%	14.28%	10.82%	15.43%	15.13%	6.24%	14.13%	10.86%	15.43%	15.06%
p31	6.88%	16.08%	10.73%	15.44%	8.50%	6.88%	15.17%	16.68%	15.44%	8.47%
p32	7.54%	10.72%	13.72%	17.22%	14.33%	6.82%	10.40%	9.12%	16.25%	13.23%
pr01	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
pr02	1.50%	1.27%	2.46%	3.11%	2.40%	1.31%	1.27%	1.78%	3.10%	2.24%
pr03	0.79%	2.35%	4.70%	5.53%	5.05%	0.79%	2.35%	4.00%	5.53%	5.04%
pr04	1.20%	5.03%	2.46%	5.96%	3.95%	1.20%	5.03%	5.52%	5.96%	3.95%
pr05	4.89%	7.93%	3.73%	5.87%	8.13%	4.89%	7.16%	4.93%	5.40%	7.62%
pr06	7.41%	11.70%	9.23%	9.27%	10.17%	7.41%	11.70%	8.10%	9.27%	10.17%
pr07	1.57%	0.31%	0.00%	1.57%	0.87%	1.57%	0.31%	1.00%	1.57%	0.87%
pr08	2.27%	3.72%	4.91%	2.26%	4.80%	2.11%	3.27%	2.14%	1.93%	4.57%
pr09	5.72%	7.22%	9.68%	8.19%	10.64%	5.67%	6.69%	5.39%	7.80%	9.31%
pr10	7.14%	6.52%	7.82%	7.75%	8.82%	6.33%	6.23%	5.45%	6.92%	8.65%
Avg	**4.71%**	**6.84%**	**6.50%**	**7.51%**	**8.09%**	**4.56%**	**6.74%**	**6.45%**	**7.23%**	**7.86%**
σ^2	**0.15%**	**0.21%**	**0.25%**	**0.27%**	**0.28%**	**0.13%**	**0.21%**	**0.22%**	**0.26%**	**0.27%**
σ	**3.82%**	**4.62%**	**5.03%**	**5.19%**	**5.27%**	**3.56%**	**4.56%**	**4.71%**	**5.06%**	**5.20%**

Note: results on the left (stoch 20…100 = DTD); results on the right (Reopt stoch 20…100 = RDF)

Appendix E

demand ordering ASC tour lengths (60-90% stochasticity)

dataset	deterministic	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	597.54	667.79	666.51	629.16	621.62	645.48	642.69	629.16	621.62
p2	1438.38	1438.38	1500.16	1515.66	1554.6	1438.38	1491.37	1515.66	1554.6
p3	628.61	712.42	656.39	692.42	656.39	712.42	656.39	692.42	656.39
p4	945.57	1041.87	1054.06	1104.26	945.57	1041.87	1054.06	1085.22	945.57
p5	2177.77	2302.8	2401.29	2495.13	2348.79	2293.01	2401.29	2485.34	2348.79
p6	972.89	1071.19	1082.19	993.04	1021.92	1071.19	1082.19	993.04	1021.92
p7	942.57	952.57	942.57	974.93	965.22	952.57	942.57	974.68	964.97
p8	2263.63	2361.25	2469.08	2331.7	2493.73	2361.25	2465.07	2331.7	2476.74
p9	916.74	934.85	934.85	934.85	934.85	934.85	934.85	934.85	934.85
p10	1762.76	1844.28	1816.85	1903.58	1893.14	1844.28	1816.85	1903.58	1893.14
p11	864.8	949.95	901.37	898.74	945.85	949.95	901.37	898.74	945.85
p12	1321.68	1519.84	1464.85	1648.84	1676.51	1467.52	1429.19	1638.72	1643.32
p13	4352.19	4509.25	4524.23	4558.09	4826.5	4499.95	4519.52	4558.09	4826.5
p14	958.58	978.58	958.58	978.58	998.58	978.58	958.58	978.58	998.58
p15	1981.3	2061.31	2121.3	2101.3	2061.3	2060.9	2121.3	2101.3	2061.3
p16	2972.67	2972.67	2972.67	3012.67	2972.67	2972.67	2972.67	3011.77	2972.67
p17	1630.77	1770.77	1650.77	1810.77	1750.77	1770.77	1650.77	1810.77	1750.77
p18	3267.95	3267.95	3287.95	3367.95	3287.95	3267.95	3287.95	3367.95	3287.95
p19	5046.64	5106.64	5286.64	5266.64	5366.64	5106.64	5282.41	5266.64	5366.64
p20	9094.58	9334.58	9194.58	9294.58	9094.58	9334.58	9194.58	9283.04	9094.58
p21	2291.9	2591.9	2611.9	2411.9	2611.9	2591.49	2611.9	2411.9	2611.9
p22	4524.47	4644.47	4984.47	4804.47	4744.47	4644.47	5203.37	4789.73	4736.17
p23	6962.6	7302.6	7302.6	7282.6	7422.61	7293.26	7298.65	7265.71	7392.79
p24	3859.19	4170.13	4295.21	4054.66	4139.19	4166.91	4295.21	4054.66	4139.19
p25	3907.61	4574.66	4539.19	4630.42	4138.55	4516.56	4539.19	4629.01	4138.55
p26	3974.63	4535.98	4295.28	4295.28	4726.22	4535.98	4252.38	4402.55	4726.22
p27	23990.77	26412.11	28395.04	26638.56	25977.73	26412.11	27837.31	26540.42	25959.56
p28	24260.25	25786.78	26889.35	26419.88	26792.63	25769.49	26889.35	26409.8	26754.8
p29	24971.93	27424.63	26154.46	26389.3	30321.55	27424.63	26154.46	26105.56	30226.12
p30	85909.78	91790.12	97451.99	94238.01	93942.44	91154.43	97376.53	93861.96	93685.77
p31	84525.69	91607.89	93526.81	90790.57	92877.23	91140.86	93526.81	90019.52	92838.16
p32	87642.98	94490.1	94687.97	96231.76	99770.54	94269.06	94336.41	96206.8	98639.72
pr01	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31
pr02	4146.13	4162.84	4174.7	4275.34	4248.27	4162.84	4174.7	4261.34	4236.22
pr03	5598.49	5675.57	5643.28	5614.15	5870.57	5675.57	5643.28	5614.15	5870.23
pr04	7360.78	7552.58	7714.24	7681.11	7705.72	7552.58	7798.55	7681.11	7705.72
pr05	7362.97	7777.4	7788.2	8049.41	7845.6	7734.95	7780.13	8042.97	7828.33
pr06	8817.96	9797.94	10024.54	9960.95	10070.05	9797.94	10024.54	9960.95	10070.05
pr07	5108.6	5108.6	5108.6	5241.79	5108.6	5108.6	5108.6	5241.79	5108.6
pr08	7581.11	8043.7	7781.63	7888.36	8055.34	8003.79	7778.47	7856.54	8033.61
pr09	11061.23	11831.32	12075.96	12289.18	12045.39	11736.46	11991.24	12257.71	11978.31
pr10	15913.03	17499.34	17098.07	17358.88	16819.57	17398.93	16998.52	17639.7	16788.97
Avg	16912.98	16909.41	12398.04024	12270.16143	12474.73	12121.02452	12374.10905	12238.15333	12431.50119
σ^2	466432863.8	542340487.7	573713915.1	555077784.8	578820549	537364594.3	571561983.2	550455634.7	572930305
σ	21597.05683	23288.2049	23952.32588	23560.08881	24058.68968	23181.12582	23907.36253	23461.79095	23935.96259

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix F

demand ordering ASC tour length % deviation stochastic vs deterministic demands (60-90% stochasticity)

dataset	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	10.52%	10.35%	5.03%	3.87%	7.43%	7.03%	5.03%	3.87%
p2	0.00%	4.12%	5.10%	7.48%	0.00%	3.55%	5.10%	7.48%
p3	11.76%	4.23%	9.22%	4.23%	11.76%	4.23%	9.22%	4.23%
p4	9.24%	10.29%	14.37%	0.00%	9.24%	10.29%	12.87%	0.00%
p5	5.43%	9.31%	12.72%	7.28%	5.03%	9.31%	12.38%	7.28%
p6	9.18%	10.10%	2.03%	4.80%	9.18%	10.10%	2.03%	4.80%
p7	1.05%	0.00%	3.32%	2.35%	1.05%	0.00%	3.29%	2.32%
p8	4.13%	8.32%	2.92%	9.23%	4.13%	8.17%	2.92%	8.60%
p9	1.94%	1.94%	1.94%	1.94%	1.94%	1.94%	1.94%	1.00%
p10	4.42%	2.98%	7.40%	6.89%	4.42%	2.98%	7.40%	6.89%
p11	8.96%	4.06%	3.78%	8.57%	8.96%	4.06%	3.78%	8.57%
p12	13.04%	9.77%	19.84%	21.16%	9.94%	7.52%	19.35%	19.57%
p13	3.48%	3.80%	4.52%	9.83%	3.28%	3.70%	4.52%	9.83%
p14	2.04%	0.00%	2.04%	4.01%	2.04%	0.00%	2.04%	4.01%
p15	3.88%	6.60%	5.71%	3.88%	3.86%	6.60%	5.71%	3.88%
p16	0.00%	0.00%	1.33%	0.00%	0.00%	0.00%	1.30%	0.00%
p17	7.91%	1.21%	9.94%	6.85%	7.91%	1.21%	9.94%	6.85%
p18	0.00%	0.61%	2.97%	0.61%	0.00%	0.61%	2.97%	0.61%
p19	1.17%	4.54%	4.18%	5.96%	1.17%	4.46%	4.18%	5.96%
p20	2.57%	1.09%	2.15%	0.00%	2.57%	1.09%	2.03%	0.00%
p21	11.57%	12.25%	4.98%	12.25%	11.56%	12.25%	4.98%	12.25%
p22	2.58%	9.23%	5.83%	4.64%	2.58%	13.05%	5.54%	4.47%
p23	4.66%	4.66%	4.39%	6.20%	4.53%	4.60%	4.17%	5.82%
p24	7.46%	10.15%	4.82%	6.76%	7.38%	10.15%	4.82%	6.76%
p25	14.58%	13.91%	15.61%	5.58%	13.48%	13.91%	15.58%	5.58%
p26	12.38%	7.47%	7.47%	15.90%	12.38%	6.53%	9.72%	15.90%
p27	9.17%	15.51%	9.94%	7.65%	9.17%	13.82%	9.61%	7.58%
p28	5.92%	9.78%	8.17%	9.45%	5.86%	9.78%	8.14%	9.32%
p29	8.94%	4.52%	5.37%	17.64%	8.94%	4.52%	4.34%	17.38%
p30	6.41%	11.84%	8.84%	8.55%	5.75%	11.78%	8.47%	8.30%
p31	7.73%	9.62%	6.90%	8.99%	7.26%	9.62%	6.10%	8.95%
p32	7.25%	7.44%	8.93%	12.16%	7.03%	7.10%	8.90%	11.15%
pr01	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
pr02	0.40%	0.68%	3.02%	2.40%	0.40%	0.68%	2.70%	2.13%
pr03	1.36%	0.79%	0.28%	4.63%	1.36%	0.79%	0.28%	4.63%
pr04	2.54%	4.58%	4.17%	4.48%	2.54%	5.61%	4.17%	4.48%
pr05	5.33%	5.46%	8.53%	6.15%	4.81%	5.36%	8.45%	5.94%
pr06	10.00%	12.04%	11.47%	12.43%	10.00%	12.04%	11.47%	12.43%
pr07	0.00%	0.00%	2.54%	0.00%	0.00%	0.00%	2.54%	0.00%
pr08	5.75%	2.58%	3.89%	5.89%	5.28%	2.54%	3.51%	5.63%
pr09	6.51%	8.40%	9.99%	8.17%	5.75%	7.76%	9.76%	7.66%
pr10	9.06%	6.93%	8.33%	5.39%	8.54%	6.39%	9.79%	5.22%
Avg	5.72%	5.98%	6.28%	6.53%	5.44%	5.84%	6.21%	6.37%
σ^2	0.17%	0.19%	0.18%	0.22%	0.15%	0.18%	0.17%	0.21%
σ	4.07%	4.33%	4.22%	4.68%	3.83%	4.25%	4.16%	4.54%

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix G

demand ordering DESC tour lengths (60-90% stochasticity)

dataset	deterministic	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	597.54	620.34	597.54	627.06	620.34	616.47	597.54	627.06	616.47
p2	1438.38	1595.20	1531.79	1467.91	1552.13	1578.33	1530.95	1467.91	1552.13
p3	628.61	692.42	699.19	692.42	712.42	692.42	699.19	692.42	712.42
p4	945.57	1048.07	1032.71	1090.95	1129.02	1048.07	1032.71	1090.95	1129.01
p5	2177.77	2283.94	2479.42	2299.74	2459.82	2274.48	2469.63	2329.76	2511.60
p6	972.89	1094.29	1128.07	972.89	1075.67	1094.29	1128.07	972.89	1075.67
p7	942.57	955.22	942.57	951.51	942.57	955.22	942.57	951.51	942.57
p8	2263.63	2470.52	2390.55	2408.40	2426.11	2455.77	2377.78	2397.89	2386.50
p9	916.74	916.74	952.80	958.93	916.74	916.74	952.80	958.93	916.74
p10	1762.76	1864.19	1883.02	1893.81	1858.62	1864.19	1883.02	1889.93	1851.74
p11	864.80	940.38	966.99	996.13	938.25	940.38	961.20	996.13	938.25
p12	1321.68	1544.28	1585.30	1830.75	1652.52	1510.15	1585.30	1807.98	1618.96
p13	4352.19	4603.88	4585.23	4706.53	4591.53	4599.17	4572.94	4695.46	4571.64
p14	958.58	958.58	998.58	1098.58	958.58	958.58	998.58	1098.58	958.58
p15	1981.30	2141.30	2141.30	2061.30	1981.30	2141.30	2141.30	2061.30	1981.30
p16	2972.67	2972.67	3052.67	3032.68	3072.67	2972.67	3052.67	3032.68	3072.67
p17	1630.77	1770.77	1730.77	1750.77	1870.77	1749.93	1730.77	1750.77	1858.02
p18	3267.95	3607.95	3487.95	3287.95	3367.95	3598.64	3487.95	3277.86	3367.95
p19	5046.64	5086.64	5226.64	5186.64	5286.64	5080.78	5220.78	5186.64	5286.64
p20	9094.58	9854.58	9114.58	9374.58	9194.59	9836.94	9113.74	9374.58	9178.68
p21	2291.90	2551.90	2691.90	2671.90	2771.90	2551.90	2677.09	2643.34	2771.90
p22	4524.47	4864.48	4884.47	5144.47	5244.48	4858.90	4873.06	5069.62	5244.47
p23	6962.60	7262.60	7502.60	7562.60	7522.60	7252.33	7502.31	7557.15	7519.36
p24	3859.19	4645.60	4791.06	4385.07	4234.13	4634.88	4671.72	4385.07	4234.13
p25	3907.61	4369.49	4254.02	4254.02	4569.18	4369.49	4254.02	4254.02	4569.18
p26	3974.63	5127.32	5136.63	5183.57	4290.10	5018.75	5028.06	5082.30	4290.10
p27	23990.77	28100.51	29482.86	27936.75	28006.97	28074.94	28688.04	27794.51	27850.55
p28	24260.25	27322.72	26902.70	27887.93	28740.32	27110.81	26902.70	27887.93	28432.29
p29	24971.93	29151.50	31187.73	28635.07	28962.35	29074.15	31116.41	28278.27	28885.00
p30	85909.78	95997.77	97422.56	99130.17	94945.59	95636.11	97422.56	97693.43	94834.87
p31	84525.69	94297.12	98772.02	98210.96	99503.15	94075.10	98772.02	97820.02	99503.15
p32	87642.98	101534.37	103111.72	101113.85	100456.39	101474.46	103111.72	100457.04	100456.39
pr01	2287.31	2382.04	2287.31	2300.02	2287.31	2382.04	2287.31	2300.02	2287.31
pr02	4146.13	4247.93	4221.58	4158.00	4246.14	4247.93	4209.53	4158.00	4232.14
pr03	5598.49	5934.49	5926.71	6051.13	5839.24	5914.10	5926.08	5968.22	5839.24
pr04	7360.78	7876.98	7925.91	7698.36	7929.08	7876.97	7986.42	7698.36	8048.68
pr05	7362.97	7928.13	8194.52	8153.11	7893.36	7890.60	8133.19	8088.94	7893.36
pr06	8817.96	10005.02	9799.45	10061.39	10040.95	10005.02	9788.23	10061.39	10040.95
pr07	5108.60	5238.65	5108.60	5241.79	5286.40	5238.65	5108.60	5241.79	5275.54
pr08	7581.11	8232.16	7844.87	7979.63	8220.80	8201.84	7832.24	7964.86	8220.80
pr09	11061.23	12263.48	12469.04	12215.43	12412.46	12158.54	12373.44	12243.95	12308.59
pr10	15913.03	17192.74	17460.28	17250.48	16996.01	17028.01	17439.73	17081.72	16794.20
Avg	16912.98	16909.41	12950.15	12855.12	12785.88	12665.72	12918.67	12771.22	12763.33
σ^2	466432863.76	599999511.95	632755884.71	626291880.93	613097577.09	597326568.49	632211316.18	615831685.71	612240833.12
σ	21597.06	24494.89	25154.64	25025.82	24760.81	24440.27	25143.81	24815.96	24743.50

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix H

demand ordering DESC tour length % deviation stochastic vs deterministic demands (60-90% stochasticity)

dataset	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	3.68%	0.00%	4.71%	3.68%	3.07%	0.00%	4.71%	3.07%
p2	9.83%	6.10%	2.01%	7.33%	8.87%	6.05%	2.01%	7.33%
p3	9.22%	10.09%	9.22%	11.76%	9.22%	10.09%	9.22%	11.76%
p4	9.78%	8.44%	13.33%	16.25%	9.78%	8.44%	13.33%	16.25%
p5	4.65%	12.17%	5.30%	11.47%	4.25%	11.82%	6.52%	13.29%
p6	11.09%	13.76%	0.00%	9.55%	11.09%	13.76%	0.00%	9.55%
p7	1.32%	0.00%	0.94%	0.00%	1.32%	0.00%	0.94%	0.00%
p8	8.37%	5.31%	6.01%	6.70%	7.82%	4.80%	5.60%	5.15%
p9	0.00%	3.78%	4.40%	0.00%	0.00%	3.78%	4.40%	1.00%
p10	5.44%	6.39%	6.92%	5.16%	5.44%	6.39%	6.73%	4.81%
p11	8.04%	10.57%	13.18%	7.83%	8.04%	10.03%	13.18%	7.83%
p12	14.41%	16.63%	27.81%	20.02%	12.48%	16.63%	26.90%	18.36%
p13	5.47%	5.08%	7.53%	5.21%	5.37%	4.83%	7.31%	4.80%
p14	0.00%	4.01%	12.74%	0.00%	0.00%	4.01%	12.74%	0.00%
p15	7.47%	7.47%	3.88%	0.00%	7.47%	7.47%	3.88%	0.00%
p16	0.00%	2.62%	1.98%	3.25%	0.00%	2.62%	1.98%	3.25%
p17	7.91%	5.78%	6.85%	12.83%	6.81%	5.78%	6.85%	12.23%
p18	9.42%	6.31%	0.61%	2.97%	9.19%	6.31%	0.30%	2.97%
p19	0.79%	3.44%	2.70%	4.54%	0.67%	3.34%	2.70%	4.54%
p20	7.71%	0.22%	2.99%	1.09%	7.55%	0.21%	2.99%	0.92%
p21	10.19%	14.86%	14.22%	17.32%	10.19%	14.39%	13.30%	17.32%
p22	6.99%	7.37%	12.05%	13.73%	6.88%	7.15%	10.75%	13.73%
p23	4.13%	7.20%	7.93%	7.44%	3.99%	7.19%	7.87%	7.40%
p24	16.93%	19.45%	11.99%	8.86%	16.74%	17.39%	11.99%	8.86%
p25	10.57%	8.14%	8.14%	14.48%	10.57%	8.14%	8.14%	14.48%
p26	22.48%	22.62%	23.32%	7.35%	20.80%	20.95%	21.79%	7.35%
p27	14.63%	18.63%	14.12%	14.34%	14.55%	16.37%	13.69%	13.86%
p28	11.21%	9.82%	13.01%	15.59%	10.51%	9.82%	13.01%	14.67%
p29	14.34%	19.93%	12.79%	13.78%	14.11%	19.75%	11.69%	13.55%
p30	10.51%	11.82%	13.34%	9.52%	10.17%	11.82%	12.06%	9.41%
p31	10.36%	14.42%	13.93%	15.05%	10.15%	14.42%	13.59%	15.05%
p32	13.68%	15.00%	13.32%	12.76%	13.63%	15.00%	12.76%	12.76%
pr01	3.98%	0.00%	0.55%	0.00%	3.98%	0.00%	0.55%	0.00%
pr02	2.40%	1.79%	0.29%	2.36%	2.40%	1.51%	0.29%	2.03%
pr03	5.66%	5.54%	7.48%	4.12%	5.34%	5.53%	6.19%	4.12%
pr04	6.55%	7.13%	4.39%	7.17%	6.55%	7.83%	4.39%	8.55%
pr05	7.13%	10.15%	9.69%	6.72%	6.69%	9.47%	8.97%	6.72%
pr06	11.86%	10.02%	12.36%	12.18%	11.86%	9.91%	12.36%	12.18%
pr07	2.48%	0.00%	2.54%	3.36%	2.48%	0.00%	2.54%	3.16%
pr08	7.91%	3.36%	4.99%	7.78%	7.57%	3.21%	4.82%	7.78%
pr09	9.80%	11.29%	9.45%	10.89%	9.03%	10.61%	9.66%	10.13%
pr10	7.44%	8.86%	7.75%	6.37%	6.55%	8.75%	6.84%	5.25%
Avg	8.00%	8.47%	8.35%	8.11%	7.69%	8.23%	8.08%	7.99%
σ^2	0.23%	0.33%	0.35%	0.28%	0.21%	0.30%	0.33%	0.28%
σ	4.76%	5.77%	5.96%	5.34%	4.58%	5.48%	5.70%	5.25%

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix I

distance ordering ASC tour lengths (60-90% stochasticity)

dataset	deterministic	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	597.54	627.06	597.54	673.23	602.01	627.06	597.54	673.23	602.01
p2	1438.38	1483.91	1540.72	1454.50	1581.92	1483.91	1529.20	1454.50	1569.96
p3	628.61	708.54	656.39	735.81	712.42	708.54	656.39	735.81	712.42
p4	945.57	1025.78	1100.73	1082.77	1067.83	1025.78	1100.73	1082.77	1104.62
p5	2177.77	2309.58	2408.07	2361.86	2462.25	2306.88	2403.46	2361.86	2452.46
p6	972.89	1033.15	1057.31	1079.33	1043.85	1033.15	1057.31	1079.33	1043.85
p7	942.57	985.84	986.51	976.51	997.41	985.84	986.51	968.43	996.99
p8	2263.63	2466.48	2428.85	2477.58	2491.17	2453.96	2405.10	2477.58	2480.66
p9	916.74	958.93	1001.12	916.74	940.82	958.93	1001.12	916.74	940.82
p10	1762.76	1762.76	1881.51	1858.22	1891.28	1762.76	1881.51	1858.22	1891.28
p11	864.80	920.49	927.07	915.49	910.73	920.49	927.07	915.49	910.73
p12	1321.68	1386.11	1357.13	1496.00	1692.45	1336.43	1349.33	1477.47	1675.39
p13	4352.19	4573.75	4536.05	4561.18	4608.82	4572.42	4524.64	4557.66	4606.36
p14	958.58	978.58	1038.58	978.58	978.58	978.58	1038.58	978.58	978.58
p15	1981.30	2021.30	2101.31	2261.31	2141.32	2021.30	2101.30	2259.80	2141.32
p16	2972.67	3092.67	3092.67	2972.67	2972.67	3090.80	3092.67	2972.67	2972.67
p17	1630.77	1670.77	1710.77	1750.77	1690.77	1670.77	1710.77	1750.77	1690.77
p18	3267.95	3367.95	3267.95	3427.95	3367.95	3357.86	3267.95	3427.95	3362.60
p19	5046.64	5046.64	5066.64	5106.64	5326.64	5046.64	5066.64	5106.64	5326.64
p20	9094.58	9354.58	9354.58	9234.58	9494.58	9354.58	9354.58	9234.58	9494.58
p21	2291.90	2491.90	2651.90	2311.90	2631.90	2491.90	2651.90	2311.90	2631.90
p22	4524.47	5004.47	4884.47	4964.48	5064.47	4993.75	4840.49	4945.93	5060.00
p23	6962.60	7142.60	7542.59	7262.60	7442.60	7142.60	7490.50	7246.12	7420.41
p24	3859.19	4585.07	4295.31	4585.07	4540.23	4485.08	4295.31	4485.08	4482.50
p25	3907.61	4382.57	4238.02	4283.10	4248.54	4382.57	4238.02	4283.10	4248.54
p26	3974.63	4526.22	4535.98	4279.28	4986.24	4526.22	4520.51	4386.55	4955.17
p27	23990.77	27227.47	28046.58	27002.82	27014.10	27078.69	27791.13	27002.82	26808.96
p28	24260.25	26076.43	26648.71	28293.41	27754.57	26059.14	26648.71	28293.41	27737.28
p29	24971.93	27064.52	28711.43	29199.06	28093.92	26987.17	28294.75	28795.47	28089.22
p30	85909.78	91871.38	97475.38	97806.09	97733.64	91871.38	97149.98	97538.38	96928.86
p31	84525.69	89783.92	92565.27	102687.54	97224.30	87556.45	90479.01	101942.39	97120.47
p32	87642.98	93860.75	97550.98	101999.08	97533.48	93725.66	97497.24	101926.83	97462.75
pr01	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31	2287.31
pr02	4146.13	4175.08	4176.54	4233.45	4278.34	4175.08	4171.18	4233.45	4263.93
pr03	5598.49	5915.21	5936.27	5790.20	5753.08	5909.40	5926.57	5784.39	5746.96
pr04	7360.78	7671.87	7440.89	7687.91	7718.86	7671.87	7440.89	7687.91	7802.64
pr05	7362.97	7462.31	8118.78	8249.28	7739.76	7456.46	8001.93	8122.43	7700.31
pr06	8817.96	9680.41	9822.01	9569.73	9942.76	9681.95	9822.01	9569.73	9942.76
pr07	5108.60	5190.04	5108.60	5190.04	5273.62	5190.04	5108.60	5190.04	5273.62
pr08	7581.11	8050.32	8002.50	8056.56	8038.59	8027.21	7991.07	8014.46	8019.98
pr09	11061.23	12092.42	11912.68	12188.01	12284.75	12145.74	11893.81	12188.01	12210.37
pr10	15913.03	16792.92	17103.54	17259.52	17147.73	16679.36	16988.23	17254.24	16996.11
Avg	16912.98	16909.41	12503.98	12893.05	12659.72	12052.90	12418.61	12851.91	12622.49
σ^2	466432863.76	533566452.51	582696263.03	644205275.85	600821486.07	524804065.59	572872506.29	639397239.39	596737027.92
σ	21597.06	23099.06	24139.10	25381.20	24511.66	22908.60	23934.76	25286.31	24428.20

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix J

distance ordering ASC tour length % deviation stochastic vs deterministic demands (60-90% stochasticity)

dataset	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	4.71%	0.00%	11.24%	0.74%	4.71%	0.00%	11.24%	0.74%
p2	3.07%	6.64%	1.11%	9.07%	3.07%	5.94%	1.11%	8.38%
p3	11.28%	4.23%	14.57%	11.76%	11.28%	4.23%	14.57%	11.76%
p4	7.82%	14.10%	12.67%	11.45%	7.82%	14.10%	12.67%	14.40%
p5	5.71%	9.56%	7.79%	11.55%	5.60%	9.39%	7.79%	11.20%
p6	5.83%	7.98%	9.86%	6.80%	5.83%	7.98%	9.86%	6.80%
p7	4.39%	4.45%	3.48%	5.50%	4.39%	4.45%	2.67%	5.46%
p8	8.22%	6.80%	8.64%	9.13%	7.76%	5.88%	8.64%	8.75%
p9	4.40%	8.43%	0.00%	2.56%	4.40%	8.43%	0.00%	1.00%
p10	0.00%	6.31%	5.14%	6.80%	0.00%	6.31%	5.14%	6.80%
p11	6.05%	6.72%	5.54%	5.04%	6.05%	6.72%	5.54%	5.04%
p12	4.65%	2.61%	11.65%	21.91%	1.10%	2.05%	10.54%	21.11%
p13	4.84%	4.05%	4.58%	5.57%	4.82%	3.81%	4.51%	5.52%
p14	2.04%	7.70%	2.04%	2.04%	2.04%	7.70%	2.04%	2.04%
p15	1.98%	5.71%	12.38%	7.47%	1.98%	5.71%	12.32%	7.47%
p16	3.88%	3.88%	0.00%	0.00%	3.82%	3.88%	0.00%	0.00%
p17	2.39%	4.68%	6.85%	3.55%	2.39%	4.68%	6.85%	3.55%
p18	2.97%	0.00%	4.67%	2.97%	2.68%	0.00%	4.67%	2.81%
p19	0.00%	0.39%	1.17%	5.26%	0.00%	0.39%	1.17%	5.26%
p20	2.78%	2.78%	1.52%	4.21%	2.78%	2.78%	1.52%	4.21%
p21	8.03%	13.58%	0.87%	12.92%	8.03%	13.58%	0.87%	12.92%
p22	9.59%	7.37%	8.86%	10.66%	9.40%	6.53%	8.52%	10.58%
p23	2.52%	7.69%	4.13%	6.45%	2.52%	7.05%	3.91%	6.17%
p24	15.83%	10.15%	15.83%	15.00%	13.95%	10.15%	13.95%	13.91%
p25	10.84%	7.80%	8.77%	8.02%	10.84%	7.80%	8.77%	8.02%
p26	12.19%	12.38%	7.12%	20.29%	12.19%	12.08%	9.39%	19.79%
p27	11.89%	14.46%	11.15%	11.19%	11.40%	13.67%	11.15%	10.51%
p28	6.96%	8.96%	14.25%	12.59%	6.90%	8.96%	14.25%	12.54%
p29	7.73%	13.02%	14.48%	11.11%	7.47%	11.74%	13.28%	11.10%
p30	6.49%	11.87%	12.16%	12.10%	6.49%	11.57%	11.92%	11.37%
p31	5.86%	8.69%	17.69%	13.06%	3.46%	6.58%	17.08%	12.97%
p32	6.62%	10.16%	14.07%	10.14%	6.49%	10.11%	14.01%	10.08%
pr01	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
pr02	0.69%	0.73%	2.06%	3.09%	0.69%	0.60%	2.06%	2.76%
pr03	5.35%	5.69%	3.31%	2.69%	5.26%	5.54%	3.21%	2.58%
pr04	4.05%	1.08%	4.26%	4.64%	4.05%	1.08%	4.26%	5.66%
pr05	1.33%	9.31%	10.74%	4.87%	1.25%	7.99%	9.35%	4.38%
pr06	8.91%	10.22%	7.86%	11.31%	8.92%	10.22%	7.86%	11.31%
pr07	1.57%	0.00%	1.57%	3.13%	1.57%	0.00%	1.57%	3.13%
pr08	5.83%	5.27%	5.90%	5.69%	5.56%	5.13%	5.41%	5.47%
pr09	8.53%	7.15%	9.24%	9.96%	8.93%	7.00%	9.24%	9.41%
pr10	5.24%	6.96%	7.80%	7.20%	4.59%	6.33%	7.77%	6.37%
Avg	5.55%	6.66%	7.31%	7.85%	5.30%	6.38%	7.16%	7.70%
σ^2	0.13%	0.16%	0.24%	0.24%	0.12%	0.15%	0.23%	0.24%
σ	3.58%	4.03%	4.90%	4.90%	3.52%	3.92%	4.75%	4.88%

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix K

distance ordering DESC tour lengths (60-90% stochasticity)

dataset	deterministic	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	597.54	626.09	672.69	644.42	627.06	627.06	626.09	670.17	644.42	627.06
p2	1438.38	1502.14	1533.91	1584.01	1553.37	1553.37	1502.14	1533.91	1583.17	1541.47
p3	628.61	628.61	648.61	648.61	692.42	692.42	628.61	648.61	648.61	692.42
p4	945.57	965.71	1092.59	987.99	1124.47	1124.47	965.71	1092.59	987.99	1124.47
p5	2177.77	2385.52	2541.36	2469.24	2360.49	2360.49	2378.82	2537.17	2469.24	2360.49
p6	972.89	1033.63	1004.51	1051.31	1004.51	1004.51	1033.63	1004.51	1051.31	1004.51
p7	942.57	942.57	952.57	986.51	966.90	966.90	942.57	952.32	986.51	956.37
p8	2263.63	2444.57	2342.51	2389.20	2549.41	2549.41	2425.27	2342.51	2389.20	2542.44
p9	916.74	958.93	916.74	916.74	1005.32	1005.32	958.93	916.74	916.74	1005.32
p10	1762.76	1833.73	1916.23	1899.79	1887.04	1887.04	1833.73	1916.23	1899.79	1887.04
p11	864.80	931.30	949.85	889.98	932.86	932.86	931.30	949.85	887.73	932.86
p12	1321.68	1612.93	1479.45	1669.22	1602.96	1602.96	1585.28	1403.17	1615.51	1557.55
p13	4352.19	4712.03	4502.78	4624.58	4524.74	4524.74	4656.01	4468.82	4621.36	4520.03
p14	958.58	958.58	958.58	958.58	958.58	958.58	958.58	958.58	958.58	958.58
p15	1981.30	2181.31	2041.30	2261.30	2101.30	2101.30	2179.44	2041.30	2261.30	2100.89
p16	2972.67	2972.67	3032.67	3032.68	2972.67	2972.67	2972.67	3032.67	3032.68	2972.67
p17	1630.77	1710.77	1830.77	1750.77	1850.77	1850.77	1704.91	1815.50	1750.77	1841.76
p18	3267.95	3327.95	3507.95	3387.95	3347.95	3347.95	3327.95	3506.50	3387.95	3347.95
p19	5046.64	5346.64	5206.64	5306.64	5126.64	5126.64	5346.64	5206.64	5304.21	5126.41
p20	9094.58	9374.58	9434.58	10494.59	9594.58	9594.58	9374.58	9434.58	10389.30	9593.74
p21	2291.90	2891.90	2491.90	2511.90	2971.90	2971.90	2891.90	2491.90	2499.43	2921.90
p22	4524.47	4964.47	5084.47	5004.47	5104.47	5104.47	4925.32	5071.97	4988.82	5100.70
p23	6962.60	7382.59	7342.60	7362.60	7562.60	7562.60	7341.59	7327.03	7362.60	7559.92
p24	3859.19	4059.19	4144.37	4279.31	4285.60	4285.60	4059.19	4144.37	4278.04	4285.60
p25	3907.61	4263.63	4764.96	4584.43	4764.01	4764.01	4263.63	4764.96	4584.43	4764.01
p26	3974.63	4366.12	4440.74	4712.53	4671.68	4671.68	4366.12	4440.74	4711.23	4778.95
p27	23990.77	27630.42	25723.60	24879.06	29235.34	29235.34	27609.23	25512.28	24879.06	29024.02
p28	24260.25	27776.89	26604.11	28124.69	25998.95	25998.95	27766.81	26600.88	28124.69	25998.95
p29	24971.93	30306.20	29235.08	28014.39	30676.70	30676.70	30215.39	29145.54	28014.39	30518.35
p30	85909.78	99436.83	97932.35	99417.67	98570.47	98570.47	98617.93	97856.89	99340.32	98570.47
p31	84525.69	94652.58	97130.49	101516.33	97899.44	97899.44	94586.71	96734.44	101516.33	97610.76
p32	87642.98	103769.56	103007.03	105975.74	103614.59	103614.59	103715.58	102650.27	105861.87	102705.34
pr01	2287.31	2300.02	2287.31	2300.02	2304.23	2304.23	2300.02	2287.31	2300.02	2304.23
pr02	4146.13	4218.94	4199.62	4205.98	4262.00	4262.00	4213.58	4199.62	4191.98	4258.22
pr03	5598.49	5751.63	5906.88	5790.84	5792.18	5792.18	5751.63	6009.83	5790.84	5792.18
pr04	7360.78	7693.48	7625.96	7682.11	7701.78	7701.78	7693.48	7625.96	7682.11	7701.78
pr05	7362.97	7942.09	7963.39	7985.49	7979.91	7979.91	7873.08	7939.06	7941.14	7905.42
pr06	8817.96	9686.91	9823.92	9863.75	9861.21	9861.21	9686.91	9823.92	9863.75	9861.21
pr07	5108.60	5190.04	5241.80	5190.04	5224.79	5224.79	5190.04	5241.80	5190.04	5224.79
pr08	7581.11	7898.67	8347.25	7867.49	8455.74	8455.74	7895.51	8321.09	7828.51	8662.43
pr09	11061.23	12019.48	12270.45	12169.47	12200.81	12200.81	11993.16	12245.22	12109.47	12201.02
pr10	15913.03	17016.65	16899.29	17491.52	17332.87	17332.87	16924.61	16878.14	17651.30	17219.90
Avg	16912.98	16909.41	12738.90	12973.43	12934.65	12934.65	12767.01	12708.23	12964.21	12896.77
σ^2	466432863.76	626541535.38	623592721.98	660198061.84	634662491.70	634662491.70	622670211.19	620014903.76	13257.54	629276670.80
σ	21597.06	25030.81	24971.84	25694.32	25192.51	25192.51	24953.36	24900.10	13535.50	25085.39

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix L

distance ordering DESC tour length % deviation stochastic vs deterministic demands (60-90% stochasticity)

dataset	det vs stoch 60	det vs stoch 70	det vs stoch 80	det vs stoch 90	det vs reopt 60	det vs reopt 70	det vs reopt 80	det vs reopt 90
p1	4.56%	11.17%	7.27%	4.71%	4.56%	10.84%	7.27%	4.71%
p2	4.24%	6.23%	9.19%	7.40%	4.24%	6.23%	9.15%	6.69%
p3	0.00%	3.08%	3.08%	9.22%	0.00%	3.08%	3.08%	9.22%
p4	2.09%	13.46%	4.29%	15.91%	2.09%	13.46%	4.29%	15.91%
p5	8.71%	14.31%	11.80%	7.74%	8.45%	14.17%	11.80%	7.74%
p6	5.88%	3.15%	7.46%	3.15%	5.88%	3.15%	7.46%	3.15%
p7	0.00%	1.05%	4.45%	2.52%	0.00%	1.02%	4.45%	1.44%
p8	7.40%	3.37%	5.26%	11.21%	6.66%	3.37%	5.26%	10.97%
p9	4.40%	0.00%	0.00%	8.81%	4.40%	0.00%	0.00%	1.00%
p10	3.87%	8.01%	7.21%	6.59%	3.87%	8.01%	7.21%	6.59%
p11	7.14%	8.95%	2.83%	7.30%	7.14%	8.95%	2.58%	7.30%
p12	18.06%	10.66%	20.82%	17.55%	16.63%	5.81%	18.19%	15.14%
p13	7.64%	3.34%	5.89%	3.81%	6.53%	2.61%	5.82%	3.71%
p14	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
p15	9.17%	2.94%	12.38%	5.71%	9.09%	2.94%	12.38%	5.69%
p16	0.00%	1.98%	1.98%	0.00%	0.00%	1.98%	1.98%	0.00%
p17	4.68%	10.92%	6.85%	11.89%	4.35%	10.18%	6.85%	11.46%
p18	1.80%	6.84%	3.54%	2.39%	1.80%	6.80%	3.54%	2.39%
p19	5.61%	3.07%	4.90%	1.56%	5.61%	3.07%	4.86%	1.56%
p20	2.99%	3.60%	13.34%	5.21%	2.99%	3.60%	12.46%	5.20%
p21	20.75%	8.03%	8.76%	22.88%	20.75%	8.03%	8.30%	21.56%
p22	8.86%	11.01%	9.59%	11.36%	8.14%	10.79%	9.31%	11.30%
p23	5.69%	5.18%	5.43%	7.93%	5.16%	4.97%	5.43%	7.90%
p24	4.93%	6.88%	9.82%	9.95%	4.93%	6.88%	9.79%	9.95%
p25	8.35%	17.99%	14.76%	17.98%	8.35%	17.99%	14.76%	17.98%
p26	8.97%	10.50%	15.66%	14.92%	8.97%	10.50%	15.63%	16.83%
p27	13.17%	6.74%	3.57%	17.94%	13.11%	5.96%	3.57%	17.34%
p28	12.66%	8.81%	13.74%	6.69%	12.63%	8.80%	13.74%	6.69%
p29	17.60%	14.58%	10.86%	18.60%	17.35%	14.32%	10.86%	18.17%
p30	13.60%	12.28%	13.59%	12.84%	12.89%	12.21%	13.52%	12.84%
p31	10.70%	12.98%	16.74%	13.66%	10.64%	12.62%	16.74%	13.41%
p32	15.54%	14.92%	17.30%	15.41%	15.50%	14.62%	17.21%	14.67%
pr01	0.55%	0.00%	0.55%	0.73%	0.55%	0.00%	0.55%	0.73%
pr02	1.73%	1.27%	1.42%	2.72%	1.60%	1.27%	1.09%	2.63%
pr03	2.66%	5.22%	3.32%	3.34%	2.66%	6.84%	3.32%	3.34%
pr04	4.32%	3.48%	4.18%	4.43%	4.32%	3.48%	4.18%	4.43%
pr05	7.29%	7.54%	7.80%	7.73%	6.48%	7.26%	7.28%	6.86%
pr06	8.97%	10.24%	10.60%	10.58%	8.97%	10.24%	10.60%	10.58%
pr07	1.57%	2.54%	1.57%	2.22%	1.57%	2.54%	1.57%	2.22%
pr08	4.02%	9.18%	3.64%	10.34%	3.98%	8.89%	3.16%	12.48%
pr09	7.97%	9.85%	9.11%	9.34%	7.77%	9.67%	8.66%	9.34%
pr10	6.49%	5.84%	9.02%	8.19%	5.98%	5.72%	9.85%	7.59%
Avg	6.78%	7.17%	7.70%	8.63%	6.59%	6.97%	7.57%	8.30%
σ^2	0.26%	0.21%	0.26%	0.32%	0.25%	0.20%	0.24%	0.32%
σ	5.09%	4.60%	5.09%	5.63%	4.97%	4.51%	4.94%	5.65%

Note: results on the left (det vs stoch 60…90 = DTD); results on the right (det vs reopt 60…90 = RDF)

Appendix M

Comparison of deterministic vs. Reoptimization After each Customer (short runtime)

dataset	deterministic	Reopt Each 60	Reopt Each 70	Reopt Each 80	Reopt Each 90
p1	597.54	704.85	629.16	726.29	627.06
p2	1438.38	1558.21	1548.89	1477.32	1532.77
p3	628.61	644.73	692.42	628.61	672.51
p4	945.57	1134.47	966.13	1039.13	1168.13
p5	2177.77	2594.15	2428.91	2342.02	2476.35
p6	972.89	1058.24	1056.96	1057.04	1029.82
p7	942.57	974.93	942.57	942.57	952.57
p8	2263.63	2546.16	2463.49	2365.46	2446.78
p9	916.74	934.85	963.18	958.93	940.82
p10	1762.76	1822.49	1866.63	1973.50	1822.49
p11	864.80	977.28	962.57	990.11	1013.61
p12	1321.68	1693.01	1815.36	1627.54	1751.06
p13	4352.19	4698.76	4591.17	4757.44	4694.79
p14	958.58	998.58	1018.58	978.58	1058.58
p15	1981.30	2061.30	2061.30	2121.30	2301.30
p16	2972.67	3052.68	3072.67	3032.67	2972.67
p17	1630.77	1790.77	1790.77	1750.77	1930.77
p18	3267.95	3267.95	3327.95	3287.95	3287.95
p19	5046.64	5226.64	5166.64	5206.64	5446.64
p20	9094.58	9194.58	9154.58	9694.58	9894.58
p21	2291.90	2471.90	2631.90	2631.90	2431.90
p22	4524.47	5064.47	5004.48	4764.47	4864.47
p23	6962.60	7382.60	7602.61	7042.60	7682.60
p24	3859.19	4170.13	4295.30	4028.90	4401.07
p25	3907.61	4484.96	4658.13	4654.66	4138.55
p26	3974.63	4420.51	4601.04	4610.75	4385.04
p27	23990.77	27853.46	27602.05	29240.27	28984.58
p28	24260.25	26092.00	27082.17	27764.58	28288.88
p29	24971.93	27315.79	29383.09	28436.03	28473.62
p30	85909.78	99413.76	101383.36	96597.15	93487.54
p31	84525.69	97274.49	105294.57	91933.40	95917.05
p32	87642.98	98116.41	99830.25	98171.88	110862.53
pr01	2287.31	2287.31	2287.31	2287.31	2287.31
pr02	4146.13	4312.25	4221.58	4248.02	4194.11
pr03	5598.49	5730.82	5986.80	5959.72	5817.18
pr04	7360.78	7644.40	7858.82	7589.75	7723.28
pr05	7362.97	7817.55	7753.02	8007.15	8112.77
pr06	8817.96	9793.83	9590.20	10246.43	9907.41
pr07	5108.60	5190.04	5190.04	5108.60	5153.21
pr08	7581.11	8159.69	8096.10	7835.21	7997.44
pr09	11061.23	11729.43	12191.81	12215.75	11916.61
pr10	15913.03	17141.02	17224.08	17156.27	17164.84
Avg	11338.02	12638.13	13006.87	12559.27	12909.84
σ^2	466432863.76	609460664.10	660328841.49	580547153.94	639043787.96
σ	21597.06	24687.26	25696.86	24094.55	25279.32

Note: break after 1min || 2000 iterations; 1200ms for TS after each Customer; random