Dynamic Traveling Salesman Problem with Drones

Master Thesis Document

Raven van Ewijk

TU Delft

Dynamic Traveling Salesman Problem with Drones

Master Thesis Document

by

Raven van Ewijk

Supervisors:	Dr.ir. J. (Joost) Ellerbroek
	Prof.dr.ir. J.M. (Jacco) Hoekstra
	Ir. C.A. (Calin) Badea
Place:	Faculty of Aerospace Engineering, Delft
Project Duration:	March, 2024 - December, 2024
Student number:	4643658

Faculty of Aerospace Engineering · Delft University of Technology

Contents

Nomenclature

List of Abbreviations

ABM	Agent Based Modelling
ATM	Air Traffic Management
CA	Continuous Approximation
DRP	Drone Routing Problem
FCS	Flight Control System
IP	Integer Programming
L&R	Launch and Retrieval
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Non-Linear Programming
MVP	Modified Voltage Potential
NLP	Non Linear Programming
NP	Nondeterministic Polynomial-time
OP	Operation
PDSTSP	Parallel Drone Scheduling TSP
T&D	Truck and Drone
TDCDP	Truck Drone Collaborative Delivery Problem
TRL	Technology Readiness Level
TSP	Travelling Salesman Problem
UAV	Unmanned Aerial Vehicle
VRP	Vehicle Routing Problem

List of Figures

List of Tables

Part I

Scientific Article

A Robust Dynamic Planning Method for Truck-Drone Delivery under Uncertainty

R.C.C. van Ewijk, C.A. Badea, J. Ellerbroek, J.M. Hoekstra
Control and Operations, Faculty of Aerospace Engineering
Delft University of Technology, The Netherlands

Abstract*—The use of drones in combination with a delivery truck can have a significant impact in improving the efficiency of last-mile delivery. Drones can be dispatched to customers from the truck, allowing the truck to continue delivering packages at the same time. This approach gives rise to the widely researched Traveling Salesman Problem with multiple Drones (TSP-mD). Numerous heuristic models have been developed to solve the problem in a near-optimal manner. However, these optimization strategies do not account for disruptions, which are common in delivery networks and can negatively impact their performance. While existing literature usually considers static models, a more dynamic approach could address these disruptions by adapting to real-time circumstances. To explore this, a dynamic method is developed in this paper for solving the TSP-mD. Its efficiency is compared to an existing static heuristic model from the literature. The comparison is performed in the BlueSky Open Air Traffic simulator, in which disruptions are introduced, such as truck delays and drone speed variations. Experiments in this environment demonstrate that the existing algorithm consistently achieves shorter mission completion times across all uncertainty settings. However, the newly developed method shows a significant improvement in performance under uncertain conditions. Therefore, the use of global optimization for the TSP-mD should be reconsidered.

***Keywords*—Traveling salesman problem with drones, dynamic, uncertainty, robustness, vehicle simulation**

I. Introduction

E-Commerce has been evolving rapidly over the past few decades, driving the need for advancements in last-mile delivery networks to support this growth. One innovative solution to enhance the efficiency of last-mile delivery is the use of drones, which can significantly streamline the process [1]. Drones can autonomously navigate, allowing them to bypass traditional road networks and traffic, ultimately enabling efficient delivery directly to customers.

The efficiency gains from drone utilization can also lead to reduced costs in last-mile logistics. Drones have the capacity to cut delivery costs by more than half compared to conventional modes of delivery [2]. Their ability to operate independently makes them particularly effective, even in urban areas [3].

Recognizing the advantages of drones in logistics, several commercial parties have developed drone technology for package delivery, such as the Google Wing and Amazon Prime Air drones, both of which have received approval from regulatory bodies [4], [5]. Although it has taken a significant amount of time to develop the technology required for drone delivery, several players in the market are demonstrating its viability. Drone delivery is now ready to be scaled [6].

Despite the multitude of upsides, relying exclusively on drones without incorporating any other vehicles into a delivery network also has its inherent downsides. Although drones can bypass road networks, they have limited delivery range and payload capacity. However, combining drones with trucks for last-mile delivery can mitigate these weaknesses. This hybrid approach leverages complementary strengths: trucks offer higher carrying capacity and extended range, allowing drones to retrieve packages directly from the truck instead of returning to the depot for each delivery. This hybrid approach therefore proves to be more efficient than traditional truck-only transport. Additionally, a last-mile delivery network with both trucks and drones would emit the least CO2 [7].

However, drone technology is not the only aspect that is required to facilitate truck-drone delivery. Logistics are also a crucial factor of the problem, especially due to its complex nature. A significant amount of research has been performed on the problem, presenting a wide range of formulations. They are mostly variations of the Traveling Salesman Problem (TSP) or the Vehicle Routing Problem (VRP), which have been shown to be nondeterministic polynomial-time ($\mathcal{NP}$)-complete [8], [9]. Given the intractability of these problems, considerable effort has gone into developing various heuristic models. These heuristics provide practical solutions that can be computed efficiently, as opposed to exact algorithms, which may take exponential time to find optimal solutions due to the problem's $\mathcal{NP}$-complete nature. Approaches such as genetic algorithms and simulated annealing have been employed to yield near-optimal solutions while balancing solution quality and computational efficiency.

Despite this focus on logistical optimization, delivery networks are inherently subject to uncertainties and are at risk of being disrupted [10]. Yet, the majority of the research on truck-drone collaborative delivery does not consider any uncertainties in their optimization models. According to Beyer et al., optimization can be viewed as a spectrum [11]. At one end lies the *simplification strategy*, where assumptions are made to reduce the problem's complexity, enabling it to be solved using conventional optimization techniques. Excluding uncertainties from the optimization model can be seen as an example of the simplification strategy on this spectrum. At the other end of the spectrum are *simulation* *optimization techniques*, which are often employed when the mathematical model is not tractable using standard mathematical procedures. Considering the uncertainties faced in real-life circumstances, simulation optimization offers a promising approach to model these complexities, motivating research that seeks to address and incorporate these unpredictable elements into optimization models for more robust solutions.

In this paper, a dynamic method is proposed that aims to improve the adaptability of the drone-assisted last-mile delivery concept to uncertainties using simulation optimization. To enable realistic conditions within the simulation, representative uncertainties that could be faced in real-life execution of such missions are incorporated, including basic traffic conditions, modified flight durations of the drone and expedited and delayed doorstep deliveries. The method is compared to an existing heuristic algorithm, which serves as a benchmark representing predetermined global optimization algorithms. The goal of the comparison is to answer the question: what is the performance difference in terms of efficiency when deploying a dynamic model instead of a model with a predetermined strategy for the truck-drone collaborative delivery problem in a realistic and dynamic environment? To answer this question, several metrics are established to quantify the efficiency of the models. This leads to the following sub questions that will assist in answering the main research question:

- How does the difference in terms of mission completion time (makespan) between the two models compare when subjecting it to different levels of uncertainty, and is there a benefit from using the dynamic method in more uncertain conditions?
- How do the models compare in terms of the distance that the drones and truck traverse?
- How is the waiting time at the pickup location of either the truck or drone affected by increasing the uncertainty?

It is not expected that the dynamic method proposed in this paper is capable of achieving superior performance in terms of makespan with respect to the existing heuristic algorithm. The heuristic algorithms are designed to get close to an optimal solution, which are difficult to match. However, under uncertain conditions, it is expected the dynamic method may prove more advantageous if these variations significantly disrupt the solutions provided by the existing algorithm.

This paper is outlined as follows: in Section II, the existing literature is described, along with a detailed explanation of the reference model. Further detailing of the problem description and its assumptions are described in Section III. The proposed dynamic method is thoroughly detailed in Section IV, while the experimental setup used to obtain the results is outlined in Section V. Then, the results are presented in Section VI. A discussion of the experiments and their results are given in Section VII. Lastly, concluding remarks are given in Section VIII.

II. Literature Review

This section presents a review of existing literature. The benchmark, reference algorithm is also explained thoroughly in this section.

A. Literature Survey

The logistics of a delivery network with a truck and drones is a relatively new concept, although recently there has been extensive research on the logistical aspects of delivery networks that deploy both trucks and drones. Consequently, this section aims to provide a comprehensive review of the existing literature, highlighting key contributions and trends. Additionally, it identifies gaps in the current knowledge and aims to explain the goal and position of this study within the literature.

Initial research on the TSP with drones was formulated with one drone assisting one truck, called the Flying Sidekick Traveling Salesman Problem (FSTSP) [12]. This led to numerous studies exploring the topic of cooperative delivery systems involving a truck and one or more drones. Due to this diversity, Zhang et al. defined a taxonomy to classify the different problems, all of which have their own nuances [13]. The Truck Drone Cooperative Delivery Problem (TDCDP) was divided into 4 distinct subproblems:

1) *Mixed delivery:* Trucks and drones can both make deliveries. Furthermore, the drones are allowed to dock on the trucks. An example of this sub-problem is the previously described FSTSP [12].
2) *Drone delivery with truck-assisting:* Drones are the only vehicles that can make deliveries, while trucks have an assisting role in driving the drones to launching locations. Drones are allowed to dock on the trucks.
3) *Parallel delivery:* Trucks and drones independently make their own deliveries.
4) *Truck delivery with drone-assisting:* Trucks are the only vehicles that can make deliveries, while drones have an assisting role in resupplying the truck with packages. Drones are allowed to dock on the trucks.

This paper addresses the situation where a truck delivers packages in cooperation with one or more drones, corresponding to the first mode of the TDCDP. In the case this formulation consists of a single truck with a single drone, it is most often called the Traveling Salesman Problem with Drone (TSP-D). Follow ups on the initial FSTSP paper [12], which was a variant of the TSP-D, were published shortly after. For instance, several different formulation methods exist, such as Integer Programming (IP) [14], Continuous Approximation (CA) [15], or Mixed Integer Linear Programming (MILP) [16].

A similar, extended problem was also introduced later, consisting of a delivery network with a single truck and multiple drones. This is commonly referred to as the Traveling Salesman Problem with multiple Drones (TSP-mD), e.g. [17]–[23]. The research also evolved to a delivery network composed of multiple trucks and multiple drones.

Some formulations consider trucks with each one drone [24], [25], while others consider that each truck individually carries multiple drones [26]–[29].

A key observation, however, is that existing research rarely considers dynamic real-life circumstances or uncertainties. The optimizations are based on deterministic inputs in most research, meaning that no uncertainties or disruptions are present in the optimization process. This can often lead to results that are susceptible to deterioration under unexpected conditions, as the solution has not been optimized to be robust to uncertainties. The existing research provides mathematically optimized (and perhaps also optimistic), yet fragile solutions due to the absence of practical feasibility and lack of responsiveness. As stated by Marczyk, optimization often prioritizes ideal conditions at the expense of robustness, leaving the solutions vulnerable to disruptions [30].

B. Reference Model Selection and Specifications

There are numerous variations in both problem formulations and solution methods, making it impractical to compare the entire body of literature to the dynamic method developed in this paper. Instead, this research focuses on comparing the performance of a single global, static optimization method with the local, dynamic method. Rather, it is the general philosophy of these models that counts in this case. Therefore, a reference heuristic algorithm is required for the comparison of the dynamic method. This reference model serves as a benchmark for the for the dynamic method, by representing the abundance of heuristic algorithms that are available for the problem in literature as outlined in Subsection II-A. The following key aspects are considered for the selection of this reference model:

1) To limit the simulation's complexity, it is desired the problem is formulated with only 1 truck;
2) With the concession of having only 1 truck, it is desired to have multiple drones on this truck, since this allows for more complex dynamics between different vehicles;
3) Since this research will focus on realistic conditions, it is desired that the model has been designed for a real city graph;
4) As will later be described in Section V, uncertainties are introduced to the simulation. Due to this unpredictability, an energy model is likely to drastically increase complexity (e.g. flat batteries due to waiting). Therefore, a range endurance model is desired.

The sole paper that satisfies all the requirements that have been posed is a paper by C. C. Murray and R. Raj, named ”The Multiple Flying Sidekicks Traveling Salesman Problem: Parcel Delivery with Multiple Drones” [18][1]. It is important to note that this paper does not fully represent all the studies discussed in Section II-A, as some models are based on significantly different assumptions. In particular, papers with vastly different formulations, assumptions, or factors like the use of multiple trucks may show considerably different responses to the introduction of a dynamic method compared to the results presented in this study.

In their paper, the authors develop a three-phased iterative heuristic, capable of producing high quality, near optimal solutions to the problem. In the first phase of the heuristic, the set of customers is partitioned into two sets. One set will be served by the truck, and the other by drones. The lower limit for the number of customers in the truck customer set is dependent on the number of drones, and will be incremented in later iterations. This limit is referred to as the Lower Truck Limit (LTL). Furthermore, phase I also produces a unique TSP tour for the truck by considering all customers in the truck customer set using a Mixed Integer Programming (MIP) model. Once the truck tour is in place, UAV customers are added and removed according to a savings metric, with the intention of reducing the completion time.

The second phase then generates sorties for the set of drone-assigned customers. These sorties are defined as the launch and recovery locations for each of the customers in that set. Furthermore, a specific drone is assigned to each of the sorties. The valid assignments that can be chosen from follow from the set P, which contains all customers i and k currently assigned to the truck's route. Crucially, this set only selects possible assignments where the range constraint of the drone is met. The subset $P^{'}$ is defined as the set that only includes the valid i, j, k (i.e. launch, customer, pickup) location combinations with consecutive i and k in the truck tour. The goal of the assignments is to find a sortie for each UAV with minimum truck and UAV waiting time.

The third and last phase then determines the timing of the operations for the truck and drones. This ensures that each operation is performed one by one, without any obstruction of other operations. This phase uses the results of phase I and II to determine the start times of each operation. Once all 3 phases are completed, the LTL is incremented by 1 and the process is repeated. Eventually, the LTL is equal to the number of customers and the problem becomes equivalent to solving a TSP. The best solution that was found is then returned, thereby returning an objective value in terms of missions completion time (makespan), a truck TSP tour, the drone sorties, and the timing of each activity. For a more thorough description of each phase of the algorithm, refer to [18].

This algorithm is used as a benchmark to represent the existing literature on the TSP-mD problem. The model is referred to as 'the reference model' from hereon. To ensure the correct functioning of the reference model, solutions available in the public repository are reproduced using the same inputs originally applied to generate these solutions. Model functioning is validated by confirming that the outputs exactly match the original results.

[1]https://github.com/optimatorlab/mFSTSP

III. Problem Description

This section covers the formal description of the Traveling Salesman Problem with multiple Drones (TSP-mD) that is considered in this paper. The problem definition is consistent with the reference model's formulation [18].

The problem at hand consists of a set of packages $\mathbb{P} = \{1, 2, ...N_p\}$, where N_p is the number of packages. Furthermore, each of these packages must be delivered by either one of the homogeneous drones or the truck. Some packages require to be delivered by the truck, because of its size, weight, whether a customer's signature is required, whether it is an inaccessible apartment, or any other reason which obstructs a drone from delivering it. The set of UAVs $\mathbb{D} = \{1, ...M\}$ are all homogeneous and can carry one package at a time, after which it must return to the truck. Furthermore, all UAVs have a range constraint, which is replenished when returning to the truck. The number of drones is described by M, which is varied between 1 and 4. The truck is assumed to have infinite carrying capacity and range in these conditions, and can thus carry all packages as well as all UAVs simultaneously. Note that the drone characteristics in this paper are not varied as in the paper of Murray and Raj [18], but are consistent with the *low range, high speed* drone instead. Its specifications can be found in Table I.

The truck is routed by adhering to a graph, which consists of nodes $\mathbb{N} = \{0, 1, ...c + 1\}$ and edges $\mathbb{E} = (i, j) \mid i, j \in \mathbb{N}$. Each edge is characterized by a certain geometry and speed limit, which may not be exceeded. The truck must adhere to the geometry of these edges and may only traverse edges (i.e. it does not travel in a Euclidean manner). It is assumed that the truck always tries to travel at the speed limit imposed by the graph's edges, although cornering and slowing down for operations result in average speeds lower than the speed limit.

At the beginning of its mission, the truck spawns at the depot. At the end of the operations, the truck along with all the drones returns to the depot. The mission is completed when the truck has arrived at the depot with all drones present in the truck, and thus with all operations completed. At each node, the truck must perform operations sequentially, which take a predefined amount of time. The exact duration for each operation can be found in Table II. The truck or drone has to stop to perform these operations, and thus these operational durations are added on top of the travel time.

TABLE I. Characteristics of the drones

Characteristic	Value
Take-off vspeed [m/s]	15.64
Cruise speed [m/s]	31.29
Landing speed [m/s]	7.82
Cruise altitude [m]	50
Range [m]	9,656

A list of the assumptions of the formulation is given below for clarity. Note that these assumptions are used by the algorithm developed by Murray and Raj [18] as well as by the dynamic method. They are as follows:

- The drones are all homogeneous and have specifications as shown in Table I;
- There is only 1 truck and M drone(s), which ranges from 1 to 4;
- The drones are only limited by its range, not energy;
- The truck cannot leave the edges of the graph;
- All respective operation durations without any uncertainties are given in Table II;
- A drone cannot be pickup up at the same location where it has been launched;
- A drone can only carry one package at a time;
- A drone's range replenishes upon returning to the truck, and can thus be launched multiple times;
- The truck does not have a range- nor a capacity constraint;
- All operations are performed sequentially, i.e. a delivery cannot be performed at the same time as a launching or retrieving a drone;
- Some customers can be served by either a drone or the truck, while others necessarily have to be served by the truck.

TABLE II. Operational times of the truck and drones

Operation	Time required [s]
Launch time	60
Recovery time	30
Package transfer to customer by drone	60
Package transfer to customer by truck	30

IV. Dynamic Method

With the purpose of enabling reactions to changes in the state of the delivery network, a method is developed with the capability of dynamically resolving changes in its surroundings. This dynamic method consists of multiple algorithms, each assigned a specific task. These include a Population-based Ant Colony Optimization (PACO) algorithm and a savings algorithm, which is represented by a MILP model. This subsection explains how the dynamic method, along with its integrated algorithms, functions.

A. PACO Algorithm and Dynamic Method Philosophy

The general philosophy of the dynamic method is as follows: the truck begins its route following a predetermined TSP tour. Since the TSP in itself is $\mathcal{NP}$-complete, it cannot be solved within polynomial-time with current exact algorithms (for further reading, refer to [8]). Therefore, a different, heuristic optimization method is implemented to solve the TSP and obtain the predetermined tour. A PACO algorithm was found to be the best global algorithm in a benchmarking experiment for TSP algorithms [31], and is therefore used for this purpose. Specifically, an implementation outlined by [32] is adhered to. In a PACO algorithm, the pheromone updates are based on the population of solutions, which distinguishes it from regular Ant Colony Optimization (ACO) algorithms [32].

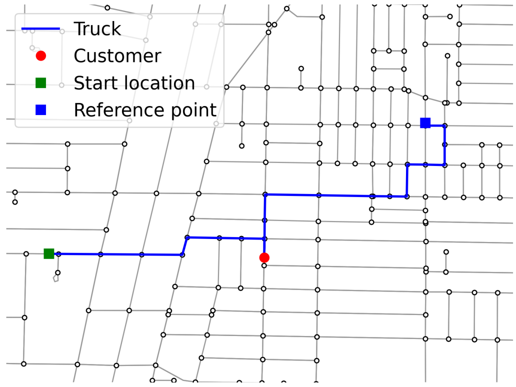
The philosophy of the dynamic method is as follows: The truck starts delivering according to the TSP determined by the PACO as if it will handle all deliveries itself. While en route, it is asserted whether serving the *next customer* in the TSP by drone results in predicted cost savings. All constraints of the problem formulation as described in Section III must still be respected. If it is predicted that savings can be achieved, the launch and pickup points for the drone are calculated, and the drone is deployed. Essentially, this resembles the greedy choice property where the choice that looks best in the current problem is made, without considering results from subproblems [33]. This process continues until all customers have been served.

B. *The Savings Algorithm and MILP Formulation*

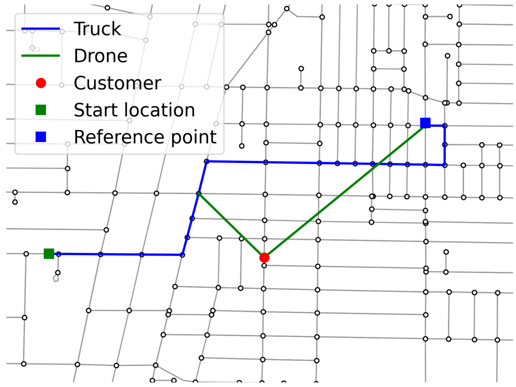
The cost savings predictions for serving the next customer by drone are achieved through the savings algorithm. This algorithm works by assessing the estimated time of arrival (ETA) at a distant, similar point in the route for both choices. That is, by taking into account the operations that have to be performed for each respective choice. This distant point is referred to as the reference point of the assessment. The main tradeoff is whether the additional time required for the launch and retrieval of a drone is sufficiently compensated for by the shortening of the route of the truck.

An example of this assessment is shown in Figure 1, where the difference in routes is shown for the choice of serving the next customer by drone. In Figure 1a, the truck first takes the shortest path to the customer, as indicated by the red dot. After serving the customer, the truck continues on its route, and passes the reference point, as marked by the blue square, at time T_{orig}. In the altered case illustrated by Figure 1b, a drone is deployed to serve the next customer. The truck now launches and retrieves the drone, which has in the meantime served the customer. Instead of driving the fastest route to the customer and only then proceeding, the truck can directly take the shortest path to the reference point. In this case, the point is passed at T_{mod}.

To assert whether savings can be made, preliminary launch- and pickup points are calculated. By determining these two locations, the impact on the truck's ETA at the reference point can be calculated, essentially comparing T_{orig} to T_{mod}. To obtain the operation locations, a Mixed Integer Linear Program (MILP) is formulated, of which the sets and parameters are formulated later in this section. The objective of the MILP is primarily to minimize the waiting time of the truck. The secondary objective of the MILP is to minimize the total mission time of the drone, which includes every aspect of its mission profile and also possible waiting time at the pickup location. Once it has been established that time savings are predicted by the algorithm, then this set of launch- and pickup location is used to deploy the drone.



(a) Truck Original Route



(b) Truck and Drone Modified Route

Figure 1. Comparison Between Truck-only vs Truck and Drone Route on the Buffalo Street Network

MILP Formulation

Sets

- $\mathbb{L}$: Set of potential launch locations
- $\mathbb{P}$: Set of potential pickup locations

Parameters

- t_i: Drone travel time from launch location i to the customer location, $\forall i \in \mathbb{L}$
- t_k: Drone travel time from the customer location to pickup location k, $\forall k \in \mathbb{P}$
- d_i: Euclidean distance from launch location i to the customer location, $\forall i \in \mathbb{L}$
- d_k: Euclidean distance from the customer location to pickup location k, $\forall k \in \mathbb{P}$
- T_i: Time at which the truck passes launch location i, $\forall i \in \mathbb{L}$
- T_k: Time at which the truck passes pickup location k, $\forall k \in \mathbb{P}$
- T_{ik}: Truck travel time from launch location i to pickup location k, $\forall i \in \mathbb{L}$, $\forall k \in \mathbb{P}$
- R: (Remaining) range of the drone
- M: Big M, large integer number

Decision Variables

- x_i: Binary variable that is 1 if the launch location i is chosen, 0 otherwise, $\forall i \in \mathbb{L}$

- y_k: Binary variable that is 1 if the pickup location k is chosen, 0 otherwise
- z_{ik}: Binary variable that is 1 if the trip from launch location i to customer location j to pickup location k is chosen, 0 otherwise, $\forall i \in \mathbb{L}$
- w^t_{ik}: Auxiliary variable representing the waiting time of the truck for the trip $i \to j \to k$, $\forall i \in \mathbb{L}$, $\forall k \in \mathbb{P}$
- w^d_{ik}: Auxiliary variable representing the waiting time of the drone for the trip $i \to j \to k$,$\forall i \in \mathbb{L}$, $\forall k \in \mathbb{P}$

Objective Function

$$\text{Min.} \quad W = \sum_{i \in L} \sum_{k \in P} \left(w^t_{ik} \cdot M + (t_i + t_k) \cdot z_{ik} + w^d_{ik} \right) \tag{1}$$

Constraints

Only one launch location can be chosen:

$$\sum_{i \in L} x_i = 1 \tag{2}$$

Only one pickup location can be chosen:

$$\sum_{k \in P} y_k = 1 \tag{3}$$

Linking constraints:

$$z_{ik} \leq x_i \quad \forall i \in L, \forall k \in P \tag{4}$$

$$z_{ik} \leq y_k \quad \forall i \in L, \forall k \in P \tag{5}$$

Flow constraint:

$$\sum_{i \in L} \sum_{k \in P} z_{ik} = 1 \quad \forall i \in \mathbb{L}, \forall k \in \mathbb{P} \tag{6}$$

Auxiliary variable Constraints:

$$w^t_{ik} \geq ((T_i + t_i + t_k) - T_k) \cdot z_{ik} \quad \forall i \in L, \forall k \in P \tag{7}$$

$$w^d_{ik} \geq (T_k - (T_i + t_i + t_k)) \cdot z_{ik} \quad \forall i \in L, \forall k \in P \tag{8}$$

Range of drone constraint:

$$R \geq d_i \cdot x_i + d_k \cdot y_k \quad \forall i \in L, \forall k \in P \tag{9}$$

Unequal launch and retrieval locations:

$$x_i + y_k \leq 1 \quad \forall i \in L, i = k \tag{10}$$

Binary and non-negative constraints:

$$x_i \in \{0, 1\} \quad \forall i \in L \tag{11}$$

$$y_k \in \{0, 1\} \quad \forall k \in P \tag{12}$$

$$z_{ik} \in \{0, 1\} \quad \forall i \in L, \forall k \in P \tag{13}$$

$$w^t_{ik} \geq 0 \quad \forall i \in L, \forall k \in P \tag{14}$$

$$w^d_{ik} \geq 0 \quad \forall i \in L, \forall k \in P \tag{15}$$

The two objectives of the optimization are captured by Equation 1. The primary objective of the MILP, which is the minimization of truck waiting time, is captured by the first term in the equation, $w^t_{ik} \cdot M$. The secondary objective, minimizing the drone's mission time, is represented by the second term, $(t_{ij} + t_{jk}) \cdot z_{ik} + w^d_{ik}$. The model's constraints are captured by Equations 2-15. Specifically, Equations 2 and 3 ensure that only 1 launch- and pickup location are selected. Decision variable z_{ik} is constrained by Equations 4 and 5, to ensure proper linkage to x_i and y_k and by Equation 6 to ensure exactly one trip is selected. The waiting time variables are linked to the selected trip z_{ik} by Equations 7 and 8. The range of the drone is constrained by Equation 9, limiting the allowable flying distance to the customer location j from launching point i and pickup point k to the drone's maximum range. To ensure consistency with the reference model, an equality between launch and retrieval location must be prohibited. This is ensured by Equation 10. Binary and non-negativity constraints are given by Equations 11-15.

The tradeoff between the choice of launching a drone or not is relatively simple in the case of a single drone. However, when the truck has more than a single drone at its disposal, the MILP can be called again for a second drone. During this process, the pickup location of previous drones can be changed to another suitable location on the truck's route, again determined by the MILP. This allows for dynamic responses to the new situation. However, to prevent a loop of changing both the launch and pickup location of a drone, a launch location is considered fixed and cannot be changed once it has been calculated for the first time. This also implies that a drone will be launched from location i regardless of any changing circumstances.

When the algorithm is called to assess the impact for a new drone, the impact on previous drone(s) is also taken into account. If any of the previous drone(s) or the truck have to be rerouted due to the launching of this new drone, the effects on the ETA at the distant point are reevaluated. This might entail that the truck now only has to travel a shorter distance, because it can skip another customer by delegating it to a drone. Potential rerouting effects on previous drone(s) are determined by again running the MILP for the previous drone(s), this time with the truck's new route and reduced range in case the drone is already performing its mission. Since the pickup location has been fixed the first time the MILP was called for the previous drone(s), only the pickup location is determined while the launch location remains fixed.

In the case of any additional *truck* waiting time, this is indicated by the outcome of the MILP. This new round of launching a drone thus also includes a penalization for truck waiting time caused by rerouting previous drone(s). In the case that savings are still possible even after reevaluating the pickup point of previous drone(s), a new drone is launched.

The same logic of recalculating the pickup location is automated even when not launching a new drone. This ensures that the truck and drone can coordinate in situations where an unexpected turn of events has occurred. The frequency of recalculation is detailed in Subsection IV-C, while the specific events that can occur are explained in Subsection V-A.

C. Method Hyperparameters

A total of three hyperparameters are identified for the model. One of these is the time interval t at which the savings algorithm is called. Naturally, a higher frequency allows the model to more actively seek for makespan reductions by launching a drone, but at the same time it makes the model slower with regards to computational time. Secondly, a timeout hyperparameter $t_{timeout}$ is introduced. This hyperparameter ensures that the model is not immediately called again after establishing that no time savings are possible by launching a new drone. Lastly, the hyperparameter t_r is introduced to specify the time interval at which the algorithm recalculates the pickup locations for all active drones without spawning a new drone. These hyperparameters remain constant throughout the experiments, with their values shown in Table III.

TABLE III. DYNAMIC METHOD HYPERPARAMETER VALUES

Parameter	t [s]	$t_{timeout}$ [s]	t_r [s]
Value	1	60	10

D. Key Differences with Reference Model

Following the definition and functioning of the dynamic method, several key distinctions from the reference model emerge. These differences are essential to consider when interpreting the results, as they assist in explaining variations between the two models. The key differences and similarities are discussed in detail in this subsection.

The primary difference between the two is that the dynamic method does not predetermine the sorties of the UAVs and the truck and drone customers. As explained in Section II-B, the reference model exactly schedules which customers are served by the truck and which are served by the UAV. Furthermore, the launch and pickup locations are also predetermined. In the dynamic method, however, none of this is decided in advance; each sortie is determined en route with the use of the savings algorithm. The logic for determining the sorties is similar: the reference model also uses a savings metric to minimize the truck and drone waiting time.

A similarity exists in the construction of the truck tour: for both models, a TSP tour is generated beforehand. However, this is done in different ways. In the reference model, an MIP (an exact solving method) is used to determine the tour, while in the dynamic method, a PACO algorithm (a heuristic solving method) is applied. Furthermore, the truck tour in the dynamic method contains all customers, since it expects the truck to solve all customers by truck. As explained before, some of these customers will in the end be served by a UAV. This thus also implies that the truck tour is not fixed in this case. For the reference model, the MIP only includes the customers that are served by the truck no matter what, resulting in a fixed truck tour.

V. EXPERIMENTAL SETUP

This section elaborates on the experimental setup within BlueSky that is used for the comparison of the two models. It covers the uncertainty implementation, variables, hypotheses, and the simulation setup of the experiments.

A. Implementation of Uncertainties

One of the aims of this paper is to include uncertainties in the simulation that could be faced in real life, to enable a comparison between the two models in these conditions. Numerous uncertainties exist in a real-life delivery network, such as traffic conditions (which is also dependent on the time of day), weather conditions, customer availability, sudden road closures, and fuel/ battery drainage. It is both infeasible and undesired to account for all uncertainties that could potentially be faced. Instead, a set of 3 uncertainty sources have been drafted to simulate the impact of uncertainties *in general*. These uncertainty aspects are:

(a) **Truck stops**: The aim of this uncertainty source is to capture e.g. a gas station stop or a traffic light. A truck stop is simulated by letting the truck come to a complete stop with a frequency of f_{stop} for a duration of t_{stop} seconds. During this time, the truck cannot perform any other operations simultaneously.

(b) **Delayed/ expedited deliveries**: To reflect the uncertainty in the delivery process, the deliveries might get delayed or expedited. Examples include situations where the customer is unavailable upon delivery or the driver takes extra time to locate the customer's package. A gamma distribution is used to model this deviation from the expected delivery time, as shown in Table II. This distribution type is often used to describe durations of processes [34], e.g. in [35]. To allow the distribution to return negative values, the mean of the Gamma distribution is shifted to $\mu_{dist} = \mu_{desired} - delay^{-}$, where $delay^{-}$ is the minimum (negative) delay. This represents the maximum amount of time a delivery can be sped up by. After generating delivery times from this distribution, the negative delay is added to it, allowing for some delivery times to be negative (i.e. expedited deliveries). The shape parameter used, $k = 3$, gives the distribution a moderate peak with a right tail, meaning that while the majority of deliveries will be centered around the expected delay (with some expedited deliveries), there is still a likelihood of a few much longer delays.

(c) **Faster/ slower drones**: Existing research generally assumes fixed time durations for traveling between two points, while flight paths of drones can be complex [36]. For example, factors like temperature are known to influence the performance of an electric drone [37]. To capture the effect of temperature and other influential factors such as wind, payload weight, malfunctions, or obstacle/ collision avoidance, the introduction of drone speeds lower or higher than the expected speed is desired. An uncertainty factor is applied on top of the expected

TABLE IV. UNCERTAINTY PARAMETER VALUES PER LEVEL

Parameter Name	Level 1	Level 2	Level 3
$f_{stop}[1/s]$	1/360	1/240	1/120
$t_{stop}[s]$	30	30	30
$\mu_{desired}[s]$	10	30	60
$delay^{-}[s]$	-10	-15	-20
$p_{spdmod}[\%]$	20	50	80
$v_{mod}[\%]$	±20	±30	±40

cruise speed of the drone, which can be found in Table I. The probability of a drone having an altered speed is given by p_{spdmod}, while the magnitude of the change is given by v_{mod}. If the speed is modified, it has a 50/50 percent chance of being slower or faster.

B. Independent Variables

Throughout the research, a number of variables are varied to investigate their impact on the dependent variables that are described in Subsection V-C. The independent variables of the experiments are the following:

- The number of customers to be served;
- The number of drones at the disposal of the truck;
- The uncertainties.

The number of customers in the experiments is varied between 10, 25, 50, and 100, while the number of drones ranges from 1 to 4. To investigate the effects of the uncertainties, different levels of uncertainty are introduced. The level of uncertainty determines the value of the aforementioned parameters of uncertainty. A total of 4 different levels are used, *0*, *1*, *2* and *3*, where *0* is the control environment without any uncertainty. The corresponding parameter values for each uncertainty level can be found in Table IV.

The inputs of both models expect there to be no uncertainty, and thus expect: 1) The truck not to stop anywhere except where there is an operation; 2) Delivery times to be equal to the ones given in Table II; 3) Drones to fly at their nominal speed as given in Table I. Only *after* an event with uncertainty has taken place, the dynamic method is able to respond to it. In the case a drone travels at an elevated or a reduced speed, the dynamic method will still continue to expect the drone to travel at its designed speed, as shown in Table I, for the remainder of its trip.

C. Dependent Variables

In Section I, the different metrics for efficiency of the two methods were outlined. The following metrics were established: Completion time (makespan), distance traveled by drone and truck, and waiting time at the pickup location.

The first dependent variable aims to capture the makespan of both methods. Since it is desired to investigate the difference between the reference model (RM) and the dynamic method (DM) in various conditions, the percentage difference between the two models is to be compared. This percentage difference is defined as the difference between the makespan of the DM w.r.t. the RM, as formulated by Equation 16.

$$\Delta MS = \frac{MS_{DM} - MS_{RM}}{MS_{RM}} \cdot 100\% \tag{16}$$

Where MS_{DM} and MS_{RM} are the makespan of the dynamic method and the makespan of the reference model, respectively. Furthermore, ΔMS is the percentage difference in makespan. The makespan values are a direct measurement of the amount of time that is required for all customers to be served.

The variables that are used to quantify the distance traveled by the drone and truck are more straightforward. Firstly, the drone distance can be assessed in distance per sortie as well as the total distance. For the truck, the distance traveled is normalized per customer that is served, such that the metric can be compared when changing the number of customers. All distanced are directly obtained from BlueSky simulation measurements, and are given in meters.

Lastly, the waiting time requires to be expressed in terms of a variable. It is defined as the absolute waiting time, i.e. the waiting time of either the truck or the drone in seconds.

D. Hypotheses

As briefly mentioned in Section I, it is not expected that the dynamic method is capable of achieving superior performance in terms of mission completion time. It is anticipated that a threshold level of uncertainty exists beyond which the performance of the representative algorithm significantly deteriorates, while the dynamic method proves better equipped to handle such conditions. The hypotheses can be summarized as follows:

- The hypothesis on mission completion time is that the relative performance of the models, as measured by ΔMS, will get closer as the uncertainty level increases;
- Furthermore, it is expected that the dynamic method is better suited to find shorter sorties due to the fact that it searches in a local manner w.r.t. the global optimization as performed by the reference model;
- Lastly, the waiting time is also expected to be lower for the dynamic method, as it is able to dynamically respond to live conditions, and thus able to avoid long (unnecessary) waiting times.

E. BlueSky- and Truck Routing Setup

The research gap identified in Section II highlights the need for an advanced simulation tool for vehicle simulation. In this study, the BlueSky Open Air Traffic simulator [38] is used to analyze and simulate the operations of truck and drone-based delivery systems. Although BlueSky's primary aim is not to simulate a truck and drone delivery network, its highly modular characteristics make it possible to modify its existing capabilities. Additionally, BlueSky is an open-source tool, making it an accessible platform for research. The modularity and accessibility are leveraged in this study to build upon and expand its functionalities, like many other studies that consider simulation of air traffic, e.g. [39], [40].

BlueSky operates with its own scripting language, TrafScript, which allows complete control of the simulation. Scenario (.scn) files can be used to chain command sequences with specific timestamps, enabling customization of scenarios to meet specific needs—in this case, a TSP-mD simulation.

As stated in Section III, a graph is required to allow the truck to be routed in the BlueSky simulation. The graphs used in this paper are obtained from OpenStreetMap [41] data, and subsequently processed using a separate module[2] to account for missing data. The OSMNx library [42] is used to access this data. Crucially, the two models use identical graphs, such that there is no discrepancy between their inputs.

The quickest route between any point A and B on this graph (these points do not necessarily overlap with nodes in the graph) is determined by the taxicab-st module[3]. To communicate these obtained routes to BlueSky, the roadroute-lib module[4] is used to obtain the corresponding TrafScript commands.

To enable simulations involving a truck and multiple drones in BlueSky, extensions to the original functionalities are used, called plugins. These plugins utilize BlueSky's modularity to extend its capabilities. These plugins can be found on Github[5]. The new functionalities that have been implemented in BlueSky mainly allow it to perform operations such as a delivery, sortie (launch) or rendezvous (pickup). Naturally, these operations are crucial for simulation of the TSP-mD.

During the simulations, data is logged directly to a .jsonl file, capturing every activity of the truck and UAVs. This includes recording each operation and the time of completion, which is defined as the moment the truck returns to the depot with all drones on board. To ensure reproducibility, a seed is set to track the randomness that influences the simulation. This randomness stems from both the uncertainties introduced in the simulation and the Population-based Ant Colony Optimization (PACO) algorithm used by the dynamic model.

F. Simulation of Reference Model

As mentioned in Subsection II-B, the reference model outputs a TSP tour of the truck, as well as drone sorties and their timings. Since these outputs are numerical values in a .CSV file, a translation between the mathematical model and the BlueSky simulation must be made. In other words, the outputs that are obtained by the reference model should be converted to BlueSky's TrafScript language. This is done by a separate module[6]. To obtain solutions, a timing table for the truck in which the time between each pair of customers is used (which is not necessarily symmetric). However, the timing tables that have been generated by the authors of the original paper are calculated using a different routing algorithm than the one used in this paper. Therefore, the timing tables are recalculated using the taxicab-st module to ensure equal inputs for the two models.

The reference algorithm has several possible settings and hyperparameters that require to be specified. These are whether or not a driver should be present for operations, whether the driver is required to perform a sortie (i.e. whether these operations can be performed simultaneously), the endurance model of the drones (e.g. constrained by energy, range, etc.) and the cutoff time for the MILP model within the heuristic. For the experiments, a driver should always be present for drone operations. Furthermore, as explained in Section III, the truck is required at the depot, and only 1 operation can be performed simultaneously. As also discussed before, the drone is considered to be restricted by a maximum range. Lastly, the cutoff time is selected to be 5 s, consistent with the assumption made by the authors of the reference model. Concluding, the assumptions and settings of the reference model coincide with the ones used for the dynamic method.

G. Selection of Instances

To ensure the number of customers can be varied and to allow for thorough comparison between the two models, a set of test instances is selected. These instances define the location and number of customers, as well as the city in which they are situated. The authors of the reference model have already provided a range of instances for testing, with customer quantities of 10, 25, 50 and 100. All Buffalo instances are selected for the experiment, consisting of a total of 40 different instances, evenly spread across customer counts 10, 25, 50, and 100. These same instances were used for the experiments in the paper by Murray and Raj [18].

VI. Results

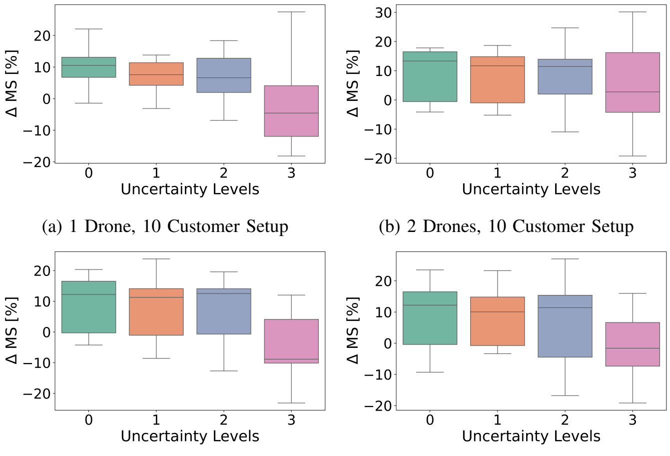
This section presents the results of the experiments, and aims to find the difference in efficiency between the reference model and the dynamic method.

A. Mission Completion Time

First, the mission completion time can be compared with the variable ΔMS. To visualize the outcomes of this variable, the measurements are grouped by the number of drones and number of customers allowing for clearer insights into individual configurations. The division of measurements results in a sample size of $n = 10$ for each group, as a consequence of the instance selection as explained in Subsection V-G. In the following figures, the different uncertainty levels are indicated by the colors green, orange, blue, and pink for levels 0, 1, 2, and 3 respectively.

The results of the measurement with 10 customers are shown in Figure 2. A high level of intra-sample variability can be observed looking at the figure. For example, the interquartile range (IQR) ranges from approximately -20%

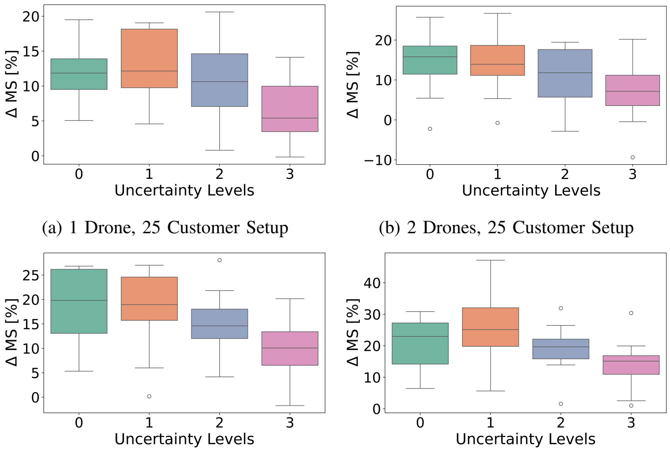
[2]https://github.com/ravenvanewijk/mFSTSP-GraphGen
[3]https://github.com/ravenvanewijk/taxicab-st
[4]https://github.com/ravenvanewijk/roadroute-lib
[5]https://github.com/ravenvanewijk/bluesky-TSP-mD/tree/TDCDP
[6]https://github.com/ravenvanewijk/mFSTSP-ScenarioGen



(a) 1 Drone, 10 Customer Setup (b) 2 Drones, 10 Customer Setup

(c) 3 Drones, 10 Customer Setup (d) 4 Drones, 10 Customer Setup

Figure 2. Δ_{perc} Measurements for 10 Customers with Varying Drone Counts



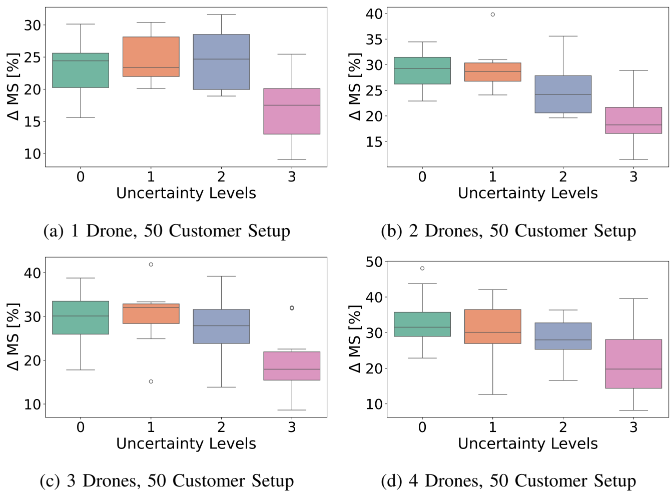
(a) 1 Drone, 25 Customer Setup (b) 2 Drones, 25 Customer Setup

(c) 3 Drones, 25 Customer Setup (d) 4 Drones, 25 Customer Setup

Figure 3. Δ_{perc} Measurements for 25 Customers with Varying Drone Counts

to +30% for the 2 drones, uncertainty level 3 setup. This variability is likely due to the fact that, for scenarios with only 10 customers, a small difference in delivery strategy can have a significant impact on the total makespan. Furthermore, the RM outperforms the DM in most cases, but occasionally, a significantly lower makespan is achieved by the DM. This can be observed in all uncertainty level 3 setups.

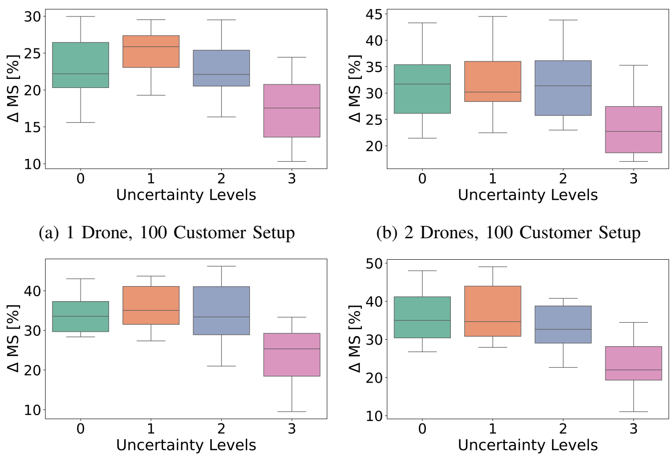
The measurements for customer count 25 are shown in Figure 3. The amount of times the RM has an inferior makespan significantly decreases, as negative ΔMS measurements are only observed in the 2 drone setups, across all uncertainty levels. The RM in this case also starts to attain some larger median differences. This is especially noticeable in the setup with 3 or 4 drones, where the median ΔMS in uncertainty level 0 is approximately 20% and 23%. For the 10 customer setups, the corresponding medians are 10% and 12%, which is significantly lower. This steep increase is a clear sign of benefits of strategic planning, showcasing that the DM is less able to leverage the additional drones that it has at its disposal than the RM. Moreover, the variability is slightly decreased for this customer count. For the 2 drones, uncertainty level 3 setup, the IQR is now reduced to approximately +0% to +20%, a significant reduction in bandwidth in comparison with the similar setup measurement with 10 customers. This bandwidth reduction can be identified across all setups. The deviations in delivery strategy thus have a less pronounced effect on the total makespan difference between the two models.



(a) 1 Drone, 50 Customer Setup (b) 2 Drones, 50 Customer Setup

(c) 3 Drones, 50 Customer Setup (d) 4 Drones, 50 Customer Setup

Figure 4. Δ_{perc} Measurements for 50 Customers with Varying Drone Counts



(a) 1 Drone, 100 Customer Setup (b) 2 Drones, 100 Customer Setup

(c) 3 Drones, 100 Customer Setup (d) 4 Drones, 100 Customer Setup

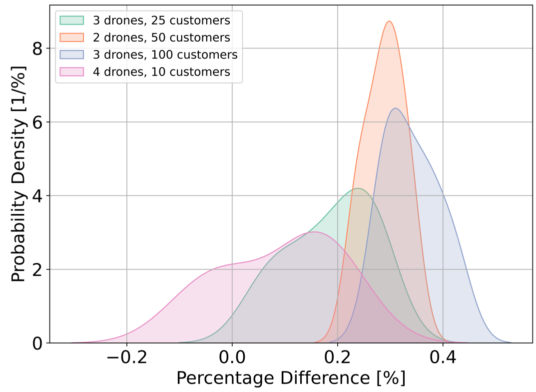
Figure 5. Δ_{perc} Measurements for 100 Customers with Varying Drone Counts

The measurements for customer count 50 are shown in Figure 4, where the trend of increasing median ΔMS is still present. For uncertainty level 0, the medians for 2, 3, and 4 drones are around 30%. A slight increase in medians is observed for 100 customers, as shown in Figure 5, with medians well above 30% in uncertainty setting 0 and drone counts 1, 2, and 3. This suggests a clear benefit of strategic pre-planning for scenarios with more customers. A similar pattern emerges with more drones, though the effect is less pronounced. The DM's localized optimization strategy cannot match the RM's performance, particularly in conditions with more customers, and to a lesser extent, with more drones.

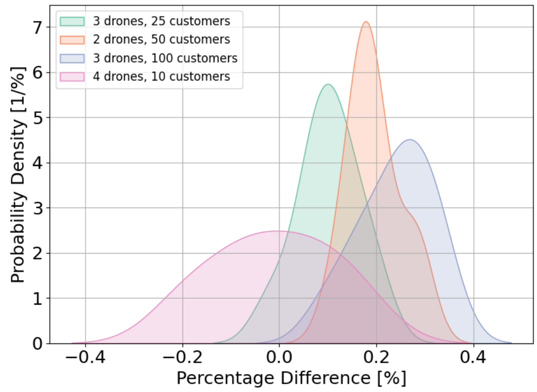
When considering all customer counts, it can be clearly seen that the DM is outperformed by the RM for most setups. This is especially true for setups where there is no or low uncertainty. The median ΔMS ranges from approximately 10% to 34%, for 1 drone, 10 customers, and 4 drones, 100 customers respectively.

Furthermore, there is almost no difference between uncertainty level 0 and level 1. This can be seen in nearly all setups. An example is the 2 drones, 50 customer setup, where the median ΔMS is approximately 30% in both cases. This trend shifts in most cases when level 2 is considered, where the median is often slightly lower. This can be observed in many cases, with some exceptions being 1 drone, 50 customers, and 2 drones, 100 customer setups, where the median ΔMS remains constant.

However, as uncertainty increases to level 3, a trend emerges where the DM catches up to the RM and, in some cases with 10 customers, even results in an improved makespan. For instance, with 1 and 3 drones, the lower bound of the IQR is approximately -5% and -7%, respectively. This trend is in general however less noticeable for these 10 customer setups. This is likely caused by the lack of circumstances to react to, i.e. the uncertainties in these setups do not have sufficient time to disrupt the RM significantly. When more customers are added, the uncertainties can propagate over time, causing these setups to be more severely impacted by the level of uncertainty. This can clearly be seen in e.g. the 2 drone, 50 customers setup, where the median gradually declines from 30% to 20%. This trend becomes clearer as the number of customers increases.



(a) KDE of Uncertainty Level 0



(b) KDE of Uncertainty Level 3

Figure 6. Sample of the Kernel Density Estimation (KDE) of Percentage Differences of the DM w.r.t the RM

Variable	p-value	Coefficient	Std. Error	95% CI
Baseline	<0.001	5.0175	0.824	[3.400, 6.634]
Unc. level 1	0.756	0.2204	0.708	[-1.169, 1.610]
Unc. level 2	0.007	-1.9021	0.708	[-3.292, -0.513]
Unc. level 3	<0.001	-8.9394	0.708	[-10.329, -7.550]
No. drones	<0.001	2.0332	0.224	[1.594, 2.473]
No. customers	<0.001	0.2443	0.007	[0.230, 0.259]

TABLE V. ANCOVA RESULTS OF ΔMS

To analyze the distribution of ΔMS, Kernel Density Estimation (KDE) is applied. A sample of the KDE results is presented in Figure 6, showing the distribution of percentage differences for a sample of four distinct setups: 3 drones with 25 customers, 2 drones with 50 customers, 3 drones with 100 customers, and 4 drones with 100 customers.

By evaluating these figures, a hypothesis is proposed suggesting that the data follows a normal distribution. This assumption is tested using the Shapiro-Wilk test, a statistical test designed to assess the normality of a dataset. If the p-value is greater than 0.05, the null hypothesis of normality is accepted; otherwise, it is rejected, indicating significant deviations from normality. From this analysis it is concluded that the p-values are higher than 0.05, and that the samples thus follow a normal distribution. Subsequently, an Analysis of Covariance (ANCOVA) is performed for the dependent variable variable ΔMS. By doing so, the effect of the uncertainty is investigated as a categorical variable, while treating the number of drones and number of customers as continuous variables (or covariates). This analysis allows isolating the individual effects of the independent variables. The results of this analysis can be found in Table V. Since the uncertainty should be treated as a categorical variable, the significance for each uncertainty level is shown.

In this table, the baseline indicates the difference between the two models in their reference conditions. This refers to the setting with 1 drone, 10 customers, and uncertainty level 0. In these conditions, the DM is expected to have approximately 5% longer completion times than the RM. The 95% confidence interval (CI) is also indicated in the table.

An analysis of the uncertainty levels is also presented. As previously noted, this variable should not be treated as continuous and is instead interpreted as a categorical variable. Therefore, separate analyses are provided for each of the three uncertainty levels. For the first level, no statistical significance is found, with a p-value of 0.756. This suggests that transitioning from uncertainty level 0 to level 1 does not have a significant impact on ΔMS. However, for the second uncertainty level, a coefficient of -1.9021 is observed with a p-value of 0.007, which is statistically significant. This indicates that switching from uncertainty level 0 to level 2 has a significant effect on the difference between the models. The same conclusion can be drawn for the third uncertainty level, where a p-value of <0.001 is observed. The coefficient in this case is -8.9394, implying a nearly 9 percentage point decrease in ΔMS when moving from uncertainty level 0 to level 3.

In addition to the analysis of uncertainty, the effects of the number of drones and the number of customers are also examined. Both variables have a significant impact on ΔMS, with a p-value of <0.001. Specifically, increasing the number of drones by one results in a 2.0332 increase in ΔMS. Similarly, increasing the number of customers by one leads to a 0.2443 increase in ΔMS. It is important to note that this increment refers to a single additional customer, even though the experiments are conducted with 10, 25, 50, or 100 customers. Nonetheless, both a larger number of drones and a greater number of customers contribute to an increased difference between the two models, favoring the RM.

Concluding, the RM has superior performance in nearly all conditions. Furthermore, w.r.t. the baseline, i.e. uncertainty level 0, 10 customers, and 1 drone, increasing either the number of customers as well as the number of drones benefits the RM. However, switching to uncertainty levels 2 or 3 improves the relative performance of the DM w.r.t the RM, with a more notable difference in the third uncertainty level.

B. Distance Traveled

In this subsection, the distance traveled by each drone is analyzed. This can be viewed as a metric of efficiency, since it assesses the ability of the model to provide convenient launch and retrieval points with minimal traveling distance.

Since each drone must take off and land exactly twice, only the horizontal distance traveled is considered. This horizontal distance is likely to be influenced by the distance between each customer; therefore, the results are grouped by customer counts of 10, 25, 50, and 100. Since the uncertainty level only affects the drone distances in the case of the DM, it is omitted from the analysis, and remains fixed at baseline level 0. The resulting visual can be observed in Figure 7. The drone's range is also indicated in this figure.

Upon analysis, it becomes evident that the reference model (RM) fails to consistently adhere to the drone's range constraint, a critical issue given the importance of this constraint in the problem. To investigate the cause of these violations, it must be recalled that the functionality of the repository was previously checked, with the model outputs successfully replicating those originally present in the repository. Subsequently, examples of violations of the range constraint are identified and verified by comparing the distances traveled by the drone in the simulation with the Euclidean distances provided in the input data table. This comparison reveals that the simulation distances closely match the input distances, often with discrepancies within 1 meter. Therefore, the issue can be attributed to the RM selecting infeasible itineraries. To further narrow down the cause of the issue, the set P' is examined to determine if the itineraries violating the range constraint are present. Such itineraries are not included in P', yet they do appear in the final solution.

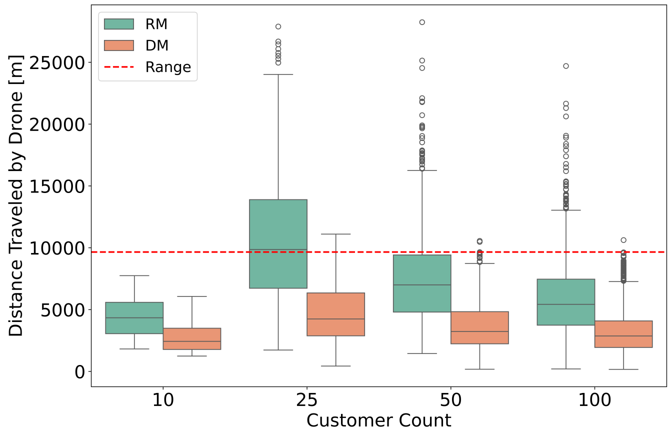


Figure 7. Distance Traveled per Drone for each Customer Count under Uncertainty Level 0

The crucial conclusion can thus be drawn that the RM does not respect the range of the drones in all cases, and in some extreme cases even traverses distances of more than 2.5 times the range of the drone. At the same time, it can also be observed that the DM in some cases exceeds the drone's range. This is likely caused by the fact that the models input parameters are given in euclidean distances. However, when rerouting the drone with a change of heading, the drones traverse a larger distance than the euclidean distance.

Furthermore, the distance traveled by a drone is lowest when only 10 customers are considered. This result is unexpected, as a lower customer density per km^2 leads to greater distances between customers, resulting in longer drone travel distances. However, in the 10-customer case, the customers are concentrated in the city center rather than spread across the larger agglomerate, offsetting the effects of the lower customer count and creating a customer density comparable to the 100-customer scenario. Therefore, the distances traveled by drone are also lower.

To avoid incorrect interpretations of this anomaly, the 10-customer case is omitted from further analysis. The remaining customer counts are analyzed through an ANCOVA, of which the results are shown in Table VI.

Variable	p-value	Coefficient	Std. Error
Model	<0.001	-3597.7432	77.218
No. customers	<0.001	-35.1886	1.260

TABLE VI. ANCOVA OF DISTANCE TRAVELED PER DRONE

As expected, the number of customers plays a significant role in distance traveled per drone, with a coefficient of -35.2. Furthermore, it becomes apparent that the model has a significant impact on the drone distance traveled with a p-value of <0.001, and a coefficient of -3598. The DM thus accomplishes significantly lower drone sorties. This can be attributed to the fact that the DM only considers local

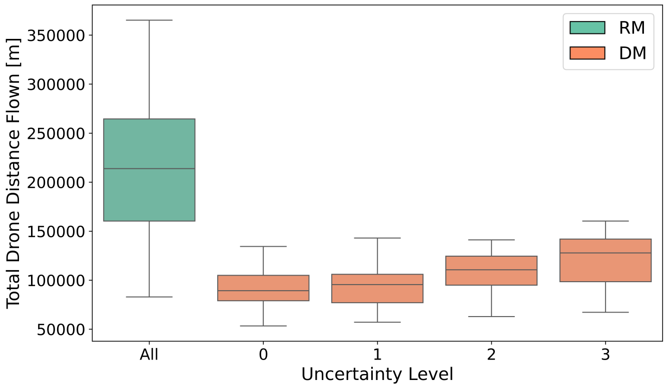


Figure 8. Total Distance Traveled by Drones for the 50 Customer Setup

improvements to the makespan instead of considering the broader picture. Its inability to serve other customers by drone than the next customer in the TSP tour means that it will not often consider customers that are located far away from the current truck position. Conversely, the RM considers a global optimization, in which all options are assessed, including distant customers. This leads to drone trips that are in general longer than the ones generated by the DM.

Further analysis of the distance traveled can be conducted by examining the total distance covered per drone. For the DM, this distance varies with each uncertainty level. In contrast, for the RM, the distance remains constant, as no deviations from the original plan are possible. The sum of distances covered by drones is shown in Figure 8. Note that the data for the RM is only shown for 1 uncertainty level, since the distance flown by drones is constant w.r.t. the uncertainty level. The figure exhibits the results only for 50 customers, since other customer counts follow a similar pattern. It can be observed that the drones cover most distance in the RM, across all uncertainty level measurements of the DM. It can also be noted that the drone distance increases as the uncertainty increases for the DM. This is likely linked with the DM finding more alternatives to serve a customer by drone, leading to a higher drone deployment level.

Lastly, the distance traveled by the truck is analyzed. It is considered a metric for the algorithm's efficiency since the truck is likely to consume most energy in the process. For this analysis, a similar approach to the one used for the distance flown by each drone is applied, focusing solely on the base uncertainty case. This is also done because of the fact that the RM's truck travel distances are not affected by the uncertainty level. Moreover, the distances that are traveled by the truck are normalized by the customer count. This way, the distance traveled per customer can be compared. The results of this analysis are shown in Figure 9.

Clearly, the truck traverses a longer distance for the DM than for the RM. The different philosophies of the two algorithms are again emphasized here: because the RM strategically plans its route beforehand, it can ensure a quick route that

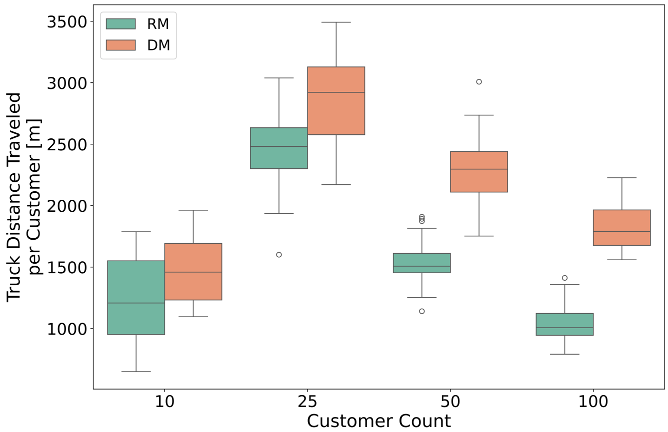


Figure 9. Distribution of Distance Traveled per Customer by the Truck

ensures no unnecessary distance is covered. However, since the DM does not account for this and does not specifically consider smarter routes in the long run, it often results in excessive driving. The assumption of fixed launching locations likely contributes significantly to these outcomes—by fixing the launch points, the truck may be forced into suboptimal routes. Adjusting the launch locations for certain drones could reduce the distance traveled and prevent this inefficiency.

Furthermore, the distance traveled by the truck per customer is lowest in the 10-customer case, for the same customer density-based reasons as explained for the drone distances.

A similar ANCOVA is performed as for the distance traveled by drone, again leaving out the 10-customer case. Results of this analysis are shown in Table VII.

Variable	p-value	Coefficient	Std. Error
Model	<0.001	545.8519	64.508
No. customers	<0.001	-4.5946	0.944

TABLE VII. ANCOVA OF DISTANCE TRAVELED PER DRONE

The analysis confirms the model has a significant impact with a p-value of <0.001 and a coefficient of 546, indicating that the DM has a longer truck distance per customer than the RM. Furthermore, the number of customers is also statistically significant, as expected, with a p-value of <0.001. Its coefficient is -4.6, indicating the truck distance per customer is reduced by 4.6 m per customer.

C. Vehicle Waiting Time at Pickup Point

This section is concluded with an analysis of the waiting time of the truck and drones at their pick up points. This comparison is conducted to evaluate the efficiency of the two models in selecting a pickup point that minimizes waiting time. The analysis is conducted for each uncertainty level, and is grouped for all customer and drone setups. This ensures a large data pool of pickup point waiting time data. The distribution of waiting times is shown in Figure 10. Note that the figure is limited to show outliers up until $1000s$, in order to preserve its readability.

Furthermore, an ANCOVA is performed on the waiting times as well, including interaction terms between the model (DM and RM) and the uncertainty level. The results are shown in Table VIII. The RM is taken as a baseline again in the analysis.

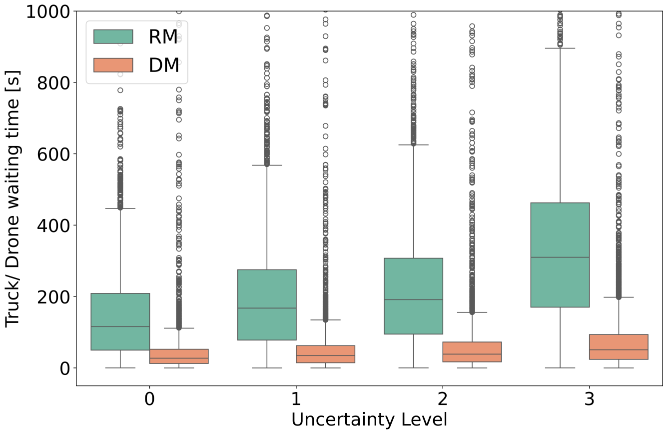


Figure 10. Waiting Time of the Truck or Drone at Pickup Point Combined for all Setups

The figure and table highlight key insights from both models. Even in the base uncertainty case, the difference in waiting time between the models is substantial. In the table, this is indicated by this coefficient of changing the model, which is -92.7. This means that without uncertainty, the waiting time of the DM is -92.7 s lower than for the RM.

As uncertainty increases, the DM shows only a minor rise in waiting time, whereas the RM experiences a more pronounced increase. The interaction terms confirm this observation, all having p-values of <0.001. The coefficients range from -43.7 to -166 as uncertainty increases, indicating the difference increases between the two models as uncertainty increases which is especially prominent for uncertainty level 3. It is very likely that this increase can be linked to the deterioration of the RM in the 3rd uncertainty level, which was highlighted in Subsection VI-A.

Although the differences between the two models are significant, it should be recalled that the DM is allowed to change its pickup point, while the RM is not allowed to do so.

On its own, the uncertainty also has a significant impact on both models, as expected. This is indicated by the p-values of <0.001 and coefficients ranging from 50 to 194, meaning that the waiting time are expected to increase 194s going from uncertainty level 0 to level 3.

Although the median waiting time of the DM is lower across all uncertainty levels, there are an unusual amount of outliers present in all cases. The explanations of this can be found in cases where the pickup point of the drone is being recalculated. If the situation occurs where the truck expects a waiting time

Variable	p-value	Coefficient	Std. Error
Baseline	<0.001	147.8749	2.581
Model	<0.001	-92.7451	3.884
Unc. Level 1	<0.001	50.0467	3.651
Unc. Level 2	<0.001	74.9891	3.651
Unc. Level 3	<0.001	193.6588	3.651
Unc. Level 1 * Model	<0.001	-43.7052	5.361
Unc. Level 2 * Model	<0.001	-63.8101	5.226
Unc. Level 3 * Model	<0.001	-166.1052	5.133

TABLE VIII. ANCOVA OF WAITING TIMES

at the current pickup location, the drone is expected to divert to a more distant point in the truck route. However, when the drone is already close to its range limit, some distant point in the route are not feasible. Therefore, alternative points later on in the route are considered as a possibility, which can be closer to the drone's current location depending on the trajectory of the truck. Since truck waiting time is heavily penalized by the MILP, the drone will decide to go to this alternative point, in favor of the pickup point where the truck would have to wait. This situation can occur even without uncertainty, as the MILP is susceptible to small amounts of expected truck waiting time, which is avoided at all costs.

VII. Discussion

This section provides a detailed discussion of the methodology that is employed and the results that are presented. These aspects should be carefully considered when interpreting the findings of the paper.

A. Dynamic Method Design Choices and Limitations

The dynamic method has many factors that influence its performance, as well as some design choices that can be a limiting factor to the performance of the model. This subsection discusses these aspects that have to be taken into account when considering the dynamic method that is presented in this paper.

One limitation of the dynamic method is that it is dependent on the results of the TSP tour that is generated prior to the simulation. This is still a static part of the algorithm and can currently not be changed in accordance with en-route circumstances. A truly dynamic model would also incorporate dynamic routing of the TSP, and would not be tied to a predetermined tour. In the case of possible road closures, last-minute changes in customers and packages, and other influences, a fully dynamic model is preferred.

Furthermore, the results highlight that the distance that the truck traverses using the dynamic method differs significantly from the distance traveled in the representative paper. The likely cause of this observation is fixing the launching locations: another element of the dynamic method that is not truly dynamic. While the general philosophy of the algorithm is being able to adapt to changing environments, there still exist a variety of conditions it cannot react to.

One way to address the issue of having fixed launching locations is by considering an optimization of launch- and pickup locations of multiple drones at once. In the proposed algorithm, the optimization using the MILP is done for one drone at a time. When more drones are available, this can result in situations where the truck is visiting launching locations of previously determined drones that could have been moved to a new location. Therefore, in the current algorithm, changing the launch location might sometimes be desirable in situations where it is currently prohibited. A significant improvement can be made by considering optimization for multiple drones simultaneously, such that a change of launch location is less likely to be needed.

Furthermore, the dynamic method has 3 hyperparameters, as mentioned previously. These hyperparameters have a significant impact on the models performance. For example, introducing intervals at which the algorithm scans for makespan improvement—i.e. as launching a new drone or rerouting another one—drastically speeds up the model. However, this approach also deteriorates its performance. It can cause the model to 'miss' certain moments where an improvement can be made, but where there is no calculation iteration running. The current hyperparameters are balanced between these two factors. However, the algorithm is still susceptible to changes in its parameters.

Additionally, as uncertainty increases, both models exhibit longer simulation durations, providing the dynamic method with more time to identify potential savings. This relationship indicates that uncertainties affect the relative significance of the hyperparameters, particularly t_s and $t_{timeout}$, leading the dynamic method to search for time savings over a longer period of time, especially after events such as truck stops. Such dynamics may enhance the performance of the dynamic method under higher uncertainty conditions.

Even without uncertainty, the hyperparameters remain crucial in determining the overall efficiency and effectiveness of the algorithm. In real-world applications, a trade-off would need to be established between computational intensity and model performance, which is contingent upon the hardware used.

Another aspect that must be considered in terms of hardware is the communication between the truck and drone(s). The truck requires communicating with all drones every t_r seconds. It was assumed that the communication between the drone and truck is permanently established and does not face a loss of signal. In reality, this might not be feasible.

B. General Limitations

Throughout the analysis, assumptions and other factors of the methodology have influenced the obtained results. Although a significant difference can be observed in the robustness of the dynamic method compared to the representative paper, the results require further investigation to verify its validity. Some noteworthy topics and takeaways can be reflected upon, and are highlighted in this subsection.

First of all, the time estimations are an important factor of the determination of the route and timing of operations for both models. Although both models use the same ETA estimation using the speed limits of the graph, this estimation is a simplification of a more complex calculation. It can be expected that an enhancement of these estimations could improve the performance of both models.

Closely linked to this are the truck dynamics. In this case, BlueSky, originally designed as an air traffic simulator, has been adapted for vehicle simulation. While the truck model has been simplified, efforts have been made to minimize the limitations inherent to this approach. It is important to acknowledge that, despite these efforts, certain aspects of truck dynamics, such as cornering behavior, may not fully align with real-world performance.

More significant limitations arise from the lack of essential driving conditions, including stopping at intersections, traffic lights, lane changes, and traffic flow. These factors can significantly influence travel times and, consequently, might harm the model's ability to (pre-)plan effectively. While an effort has been made with the truck stopping uncertainty, there is still a significant simplification factor present within this model. Integrating real traffic simulations would improve the ability of the model to produce realistic results, as done in e.g. [43].

Another computational limitation can be found in the uncertainties. The uncertainties introduced in the simulation are simplified and are modeled to induce deviations from the nominal conditions. The choice of the uncertainty sources is arbitrary, and might have a significant impact on the results. As became evident in the Section VI, the difference between uncertainty settings 2 and 3 is significant. The framework of the uncertainties likely has an impact on this outcome.

There are further limitations of the models' ability to effectively capture the real-world that must be acknowledged. In both models, it is assumed that the drone can fly from any location to any other arbitrary destination within its range. This neglects any buildings, no fly zones and other obstacles in its way. Neither model takes this factor into account. While this does not directly impact the performance of the models, it does harm their real-world applicability.

Another simplification is found in the endurance model used. Due to the complexities that an energy constraint would introduce, it is assumed that the drones are limited by a range constraint. While this approach simplifies scenarios, it is important to note that, in practice, the endurance of a drone is often the limiting factor for its range. Most energy consumed by a delivery drone is used to counteract gravity while hovering, which is the primary mechanism for lift. Consequently, the time a drone spends in the air remains relatively constant, whether it is hovering or flying. Although using a range constraint is more practical for model input, a better representation of a drone's endurance is necessary for more accurate modeling.

The air traffic conditions are also highly simplified in some cases. The model neglects any other air traffic that might be encountered during the delivery process. Furthermore, separation between drones at launch- and pickup locations is also not accounted for, which might be problematic when drones attempt to hover at the pickup location. These aspects might slow down the drones during their mission to the customer, or might cause delays at launch- and pickup locations when a large number of operations are performed at the same location.

C. Implications of the Results

While significant differences can be observed between the reference model and the dynamic method, the results should be interpreted carefully. In its core, the dynamic method is an algorithm with one basic rule: launch a drone whenever time savings are predicted. This limited the results it is capable of producing, which can likely be improved by enhancing the algorithm's philosophy and structure. Nevertheless, it provides an entry point for research on dynamically tackling this problem.

It is essential to recognize this fundamentally straightforward nature, whereas the reference model is intricate and highly detailed. The logic of the dynamic method is basic on a greedy algorithm [33], in which a local improvement is accepted no matter what, disregarding the global consequences of that choice. It is therefore noteworthy that this relatively basic, naive algorithm manages to get relatively close to the performances achieved by the reference model when disruptions are introduced to the simulation.

This observation raises the question whether it is advisable to continue using strategic, centrally optimized models. Ultimately, what is the value of employing such a model if its theoretical performance is never realized in practice, especially when a dynamic model can potentially match its effectiveness by adapting to real-time conditions without the need for pre-planning? To answer this question, the potential of dynamic methods designed for these applications should be further discovered. If a more sophisticated model can match or even outperform a strategically planned model, it would be advisable to deploy these dynamic models.

At the same time, the advanced models of existing research such as [18] and [21] should not be disregarded in this comparison. It is evident that the strategic planning is still rewarded with a significantly shorter truck route and lower mission completion times. In the base case, the dynamic method is outperformed by the reference model, partly since the truck obtains a more efficient route. The dynamic method causes the model to allocate significantly longer routes to the truck, which are often unnecessary. This highlights the effectiveness of strategic pre-planning.

Therefore, a possible way forward for these strategic models would be to integrate dynamic and strategic planning within a greater delivery management module. Developing a robust scheduling approach for the strategic tour is also a feasible option. However, the costs associated with implementing such a robust plan need to be carefully examined. In the end, uncertainties will remain unpredictable in both effect and magnitude, and accounting for them beforehand poses an interesting question.

VIII. Conclusion and Future Research

The aim of this paper is to investigate the potential efficiency benefits of a dynamic method for solving the Traveling Salesman Problem with drones. The dynamic method's objective is to enable real-time decision-making in the face of uncertainty. This algorithm is compared to an existing model from the literature, evaluating performance based on key efficiency metrics. These tests are conducted using BlueSky, under varying levels of uncertainty and with different numbers of drones and customers.

The results indicate that, in terms of mission completion time, the dynamic method is outperformed by the existing model across all uncertainty levels, drone counts, and customer configurations. However, the performance gap narrows as uncertainty increases, suggesting that the dynamic method is more robust in unpredictable scenarios. Next to this, the median distance covered by a drone is significantly shorter for the dynamic method. Conversely, the distance covered by the truck is found to be significantly longer. This indicates that the dynamic method is able to efficiently deploy drones for package delivery, but this comes at the cost of a longer truck route. Further benefits of the dynamic method are found in the decreased truck or drone waiting time at the pickup location. Although there are efficiency gains attainable with a dynamic method, in its current form the strategic pre-planning algorithms remain superior.

While the dynamic approach is less effective overall than the current pre-planning algorithms, it is less affected by uncertainties when they are increased. The topic holds significant potential for future research with numerous opportunities for further exploration. One possible research direction should focus on enhancing the dynamic method, e.g. by considering multiple drones in the MILP simultaneously or tuning the hyperparameters. Potentially, it could be combined with (simplified) pre-planning algorithms to increase its flexibility, while retaining the advantages of pre-planning. Additional realism should also be incorporated in the simulation by including aspects such as collision avoidance with other air traffic, separation at the launch- and pickup points, avoidance of obstacles and no fly zones. Furthermore, more advanced sources of uncertainty with smaller increments should be further developed. An analysis of dominant uncertainty sources should also be made to investigate which exact sources significantly impact the models' performance.

References

[1] R. Mangiaracina, A. Perego, A. Seghezzi, and A. Tumino, "Innovative solutions to increase last-mile delivery efficiency in b2c e-commerce:

a literature review," *International Journal of Physical Distribution & Logistics Management*, vol. 49, no. 9, pp. 901–920, 2019.

[2] A. Raghunatha, E. Lindkvist, P. Thollander, E. Hansson, and G. Jonsson, "Critical assessment of emissions, costs, and time for last-mile goods delivery by drones versus trucks," *Scientific Reports*, vol. 13, no. 1, p. 11814, 2023.

[3] G. Brunner, B. Szebedy, S. Tanner, and R. Wattenhofer, "The urban last mile problem: Autonomous drone delivery to your balcony," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 1005–1012.

[4] The Verge, "Google's alphabet wing gains faa approval for commercial drone deliveries," https://bitly.cx/74BL, April 2019, accessed: 2024-04-04.

[5] CNBC, "Amazon prime air drone delivery fleet gets faa approval," https://www.cnbc.com/2020/08/31/amazon-prime-now-drone-delivery-fleet-gets-faa-approval.html, August 2020, accessed: 2024-04-04.

[6] T. Papandreou, "Beyond hype: Drone delivery takes flight in 2024," *Forbes*, 2024, accessed: 2024-06-17. [Online]. Available: https://www.forbes.com/sites/timothypapandreou/2024/03/13/beyond-hype-drone-delivery-takes-flight-in-2024/

[7] A. Goodchild and J. Toy, "Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing co2 emissions in the delivery service industry," *Transportation Research Part D: Transport and Environment*, vol. 61, pp. 58–67, 2018.

[8] K. L. Hoffman, M. Padberg, G. Rinaldi *et al.*, "Traveling salesman problem," *Encyclopedia of operations research and management science*, vol. 1, pp. 1573–1578, 2013.

[9] P. Toth and D. Vigo, *The vehicle routing problem*. SIAM, 2002.

[10] A. M. Dettenbach and S. Ubber, "Managing disruptions in last mile distribution," in *2015 48th Hawaii International Conference on System Sciences*, 2015, pp. 1078–1087.

[11] H.-G. Beyer and B. Sendhoff, "Robust optimization – a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33, pp. 3190–3218, 2007.

[12] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[13] R. Zhang, L. Dou, B. Xin, C. Chen, F. Deng, and J. Chen, "A review on the truck and drone cooperative delivery problem," *Unmanned Systems*, vol. 12, no. 4, pp. 1–25, 2024.

[14] N. Agatz, P. Bouman, and M. Schmidt, "Optimization approaches for the traveling salesman problem with drone," *SSRN Electronic Journal*, 01 2015.

[15] J. G. Carlsson and S. Song, "Coordinated logistics with a truck and a drone," *Management Science*, vol. 64, no. 9, pp. 4052–4069, 2017.

[16] P. Bouman, N. Agatz, and M. Schmidt, "Dynamic programming approaches for the traveling salesman problem with drone," *SSRN Electronic Journal*, 01 2017.

[17] S. Ferrandez, T. Harbison, T. Weber, R. Sturges, and R. Rich, "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm," *Journal of Industrial Engineering and Management*, vol. 9, p. 374, 04 2016.

[18] C. C. Murray and R. Raj, "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones," *Transportation Research Part C: Emerging Technologies*, vol. 110, pp. 368–398, 2020.

[19] Z. Luo, M. Poon, Z. Zhang, Z. Liu, and A. Lim, "The multi-visit traveling salesman problem with multi-drones," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103172, 2021.

[20] P. A. Tu, N. T. Dat, and P. Q. Dung, "Traveling salesman problem with multiple drones," in *Proceedings of the 9th International Symposium on Information and Communication Technology*, ser. SoICT '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 46–53.

[21] S. Poikonen and B. Golden, "Multi-visit drone routing problem," *Computers & Operations Research*, vol. 113, p. 104802, 2020.

[22] M. R. Salama and S. Srinivas, "Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites," *Transportation Research Part E: Logistics and Transportation Review*, vol. 164, p. 102788, 2022.

[23] B. Mahmoudi and K. Eshghi, "Energy-constrained multi-visit tsp with multiple drones considering non-customer rendezvous locations," *Expert Systems with Applications*, vol. 210, p. 118479, 2022.

[24] B. Tong, J. Wang, X. Wang, F. Zhou, X. Mao, and W. Zheng, "Optimal route planning for truck–drone delivery using variable neighborhood tabu search algorithm," *Applied Sciences*, vol. 12, no. 1, 2022.

[25] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5772–5782, 2021.

[26] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. Tanchoco, and P. A. Brunese, "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach," *Computers & Industrial Engineering*, vol. 129, pp. 14–30, 2019.

[27] F. Tamke and U. Buscher, "A branch-and-cut algorithm for the vehicle routing problem with drones," *Transportation Research Part B: Methodological*, vol. 144, pp. 174–203, 2021.

[28] K. Kloster, M. Moeini, D. Vigo, and O. Wendt, "The multiple traveling salesman problem in presence of drone- and robot-supported packet stations," *European Journal of Operational Research*, vol. 305, no. 2, pp. 630–643, Mar. 2023.

[29] P. Kitjacharoenchai, B.-C. Min, and S. Lee, "Two echelon vehicle routing problem with drones in last mile delivery," *International Journal of Production Economics*, vol. 225, p. 107598, 2020.

[30] J. Marczyk, "Stochastic multidisciplinary improvement - beyond optimization," *8th Symposium on Multidisciplinary Analysis and Optimization*, vol. 4929, 09 2000.

[31] T. Weise, R. Chiong, J. Laessig, K. Tang, S. Tsutsui, W. Chen, Z. Michalewicz, and X. Yao, "Benchmarking optimization algorithms: An open source framework for the traveling salesman problem," *Computational Intelligence Magazine, IEEE*, vol. 9, pp. 40–52, 08 2014.

[32] S. Oliveira, M. S. Hussin, T. Stützle, A. Roli, and M. Dorigo, "A detailed analysis of the population-based ant colony optimization algorithm for the tsp and the qap," pp. 13–14, 07 2011.

[33] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2022.

[34] A. Ranjbari, C. Diehl, G. Dalla Chiara, and A. Goodchild, "Do parcel lockers reduce delivery times? evidence from the field," *Transportation Research Part E: Logistics and Transportation Review*, vol. 172, p. 103070, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1366554523000583

[35] W. Michiels, I. Morărescu, and S. Niculescu, "Consensus problems with distributed delays, with application to traffic flow models," *SIAM J. Control. Optim.*, vol. 48, pp. 77–101, 2009.

[36] S. Poikonen and J. Campbell, "Future directions in drone routing research," *Networks*, vol. 77, 09 2020.

[37] S. J. Kim, G. J. Lim, and J. Cho, "Drone flight scheduling under uncertainty on battery duration and air temperature," *Computers & Industrial Engineering*, vol. 117, pp. 291–302, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S036083521830041X

[38] J. M. Hoekstra and J. Ellerbroek, "Bluesky atc simulator project: an open data and open source approach," in *Proceedings of the 7th international conference on research in air transportation*, vol. 131. FAA/Eurocontrol Washington, DC, USA, 2016, p. 132.

[39] E. Sunil, J. Ellerbroek, J. Hoekstra, A. Vidosavljevic, M. Arntzen, F. Bussink, and D. Nieuwenhuisen, "Analysis of airspace structure and capacity for decentralized separation using fast-time simulations," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 1, pp. 38–51, 2017.

[40] M. J. Ribeiro, J. Ellerbroek, and J. Hoekstra, "Analysis of conflict resolution methods for manned and unmanned aviation using fast-time simulations," in *Proceedings of the 9th SESAR Innovation Days*. Athens, Greece: SESAR, 2019.

[41] OpenStreetMap contributors, "Planet dump retrieved from https://planet.osm.org ," https://www.openstreetmap.org, 2017.

[42] G. Boeing, "Modeling and analyzing urban networks and amenities with osmnx," 2024.

[43] R. F. Benekohal and J. Treiterer, "Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions," *Transportation Research Record*, pp. 99–111, 1988.

Part II

Literature Review & Research Definition

1
General Introduction

During the digital age, e-commerce has taken a significant role for retailers. This can be attributed to advancements that are made in technology, logistics, payments and trust in e-commerce, and is fueled by a generation of consumers who want greater convenience [1]. According to a report of Boston Consulting Group (BCG), e-commerce will account for 41% of the retail sales in 2027 [2]. Consequently, this increasing demand must be matched by delivery and fulfillment networks capable of accommodating it. As a response to this growing demand for online retailing, Amazon completed its first commercial drone delivery in 2016 [3], after CEO Jeff Bezos assured the public "As soon as Amazon can work out the regulations and figure out how to prevent your packages from being dropped on your head from above, there will be a fleet of shipping drones taking the sky" in 2013 [4].

The need for enhanced delivery options is further backed up by a report of leading consulting firm KPMG, who found that 43% of consumers cited this as top factor for deciding where to buy. Furthermore, according to Ragunatha et al. [5], the usage of drones can cut more than half the delivery costs when compared to other vehicles. Because of this, Amazon is not the only one trying to enter the drone delivery market, with also Google, Walmart, UPS and DHL having the same intentions [6, 7, 8, 9]. The use of drones in deliveries has also been approved by the Federal Aviation Administration (FAA), with Google's company Wing receiving the first approval in 2019, and Amazon also receiving approval shortly after [10, 11]. Amazon now even announced its new Prime Air drone that can deliver packages of up to five pounds, and is equipped with 'sense and avoid' technology that can be used to avoid obstacles such as people and property [12].

As pointed out by Agatz et al. [13], drone only delivery also puts forward some inherent downsides. The size of the drone limits the payload the drone is able to carry, and because it is battery powered it also has a limited range. The former also implies the drone would have to return to the depot every time a delivery has been completed, which is an inefficient process. Trucks on the other hand have high carrying capacity and also have a large service range. Similarly, however, truck only delivery has some downsides as well, for instance the environmental footprint and the constraints of a road network. Therefore, truck- or drone-only delivery networks have inherent downsides, linked to the nature of the system. Combining a truck and drone delivery network solves a large portion of these downsides, utilizing the complementary strengths of trucks and drones. The advantages and disadvantages of the delivery network are summarized in Table 1.1.

The potential for drones to work alongside trucks initiated research into both the technical as well as the operational aspects of this type of delivery network. In terms of this operational research, Murray and Chu [14] first investigated the use of cooperative delivery with trucks and drones. Since then, numerous mathematical models have been developed, exploring a broad spectrum of possibilities of truck-drone collaboration in delivery networks. The models presented in these papers are mostly variations of the Traveling Salesman Problem (TSP) or the Vehicle Routing Problem (VRP), which has been shown to be nondeterministic polynomial-time (NP)-complete [15, 16]. This means both problems belong to the NP and NP-hard class. NP, NP-hard, and NP-complete are all complexity classes used to classify the computational problems according to their time- and memory intensity. Solutions to the NP-complete problems can be easily verified, but they cannot be solved in polynomial time, since the solving time required explodes for larger instances [17]. Due to this nature of the problem, many of these papers employ heuristic algorithms to find near-optimal solutions within acceptable time.

Table 1.1: Advantages and disadvantages of delivery types

Delivery Type	Advantages	Disadvantages
Truck Only	+ High carrying capacity	- High operating costs
	+ Easy delivery: No need to ascend and descend	- High environmental footprint
	+ Large operating range	- Constrained by roads and road traffic
Drone Only	+ Low operating costs	- Low carrying capacity
	+ Low environmental footprint	- Slower delivery: Need to ascend and descend
	+ No road constraints	- Limited operating range
Truck and Drone	+ High carrying capacity	- Additional procedures
	+ Large operating range	
	+ Limited road constraints	
	+ Limited environmental footprint	

This project revolves around a simulation of a delivery network with a truck and drones. Within a simulation uncertainties can be introduced in a simulation, allowing for more representative real-world conditions to be mimicked. This is a crucial contribution to make the optimization of such problems more realistic and robust. Furthermore, in terms of technological maturity, a proof of concept in a simulated environment contributes to further development of the technology. This is commonly referred to as Technology Readiness Level (TRL) [18]. This project concerns a simulation of the truck-drone delivery network, in which for instance separation between drones and delays of operations or traffic jams can be investigated. These aspects are overlooked by current studies, where the aim is primarily on efficient routing in a static environment rather than on proposing a dynamic and feasible algorithm. The aim of this research is to accelerate the technology that facilitates truck drone delivery and to make it more employable in a real-world scenario.

The approach involves simulating the solutions of an existing heuristic model from literature. Furthermore, a novel, dynamic algorithm will be developed as well, which will be compared to the existing model in the simulation in various conditions. These conditions also include uncertainties, which are used to more closely capture the environment that can be expected during real-life operations. The simulations are conducted in BlueSky [19], an open-source air traffic simulator developed to make research more comparable. This makes it a suitable tool for the project, as it can also serve as a foundation for further analysis and research.

This report is structured as follows: Chapter 2 thoroughly discusses the existing literature on truck-drone network operations. This chapter is concluded with identifying the gap in existing research. This research gap is subsequently addressed in Chapter 3. The planning for the project is presented in Chapter 4. The code structure that is used for the research is explained in more detail in Chapter 5. Lastly, some additional results can be found in Chapter 6.

2
Literature Review

This chapter is a review of the research that has been conducted on existing literature is formulated. First, the existing literature is analyzed in Section 2.1. Conclusions are subsequently drawn from this analysis in Section 2.2

2.1. Analysis of Existing Literature

This section elaborates on the existing literature that has been reviewed. It is further divided into 3 distinct subsections: firstly, the existing literature on the Traveling Salesman Problem with drones is discussed in Subsection 2.1.1. Then, reviewed literature regarding conflict avoidance and collision free path planning is discussed in Subsection 2.1.2. Lastly, literature regarding the Traveling Salesman Problem without any drones, but with dynamically changing conditions is discussed in Subsection 2.1.3.

2.1.1. Truck and Drone Delivery

This subsection consists of a literature review on the truck and drone (T&D) cooperative delivery problem (TDCDP). In such a problem, delivery trucks and drones (also commonly called Unmanned Aerial Vehicles or UAVs) work in tandem to deliver packages. The problem was formally introduced in 2015 by a paper by Murray and Chu [14], where a single T&D cooperate to deliver a set of packages. The problem was formulated as a Mixed Integer Linear Program (MILP) problem and referred to as the Flying Sidekick Traveling Salesman Problem (FSTSP). Another distinct version of the TDCDP was formulated, called the parallel drone scheduling TSP (PDSTSP), where trucks and drones deliver packages independently. Following the initial article, research on the TDCDP has rapidly evolved. Murray and Chu's work [14] and the bright outlook of truck drone delivery as highlighted in Chapter 1 sparked a wave of diverse studies exploring numerous aspects of this problem. This has significantly broadened the scope of investigation in the field. This diversity gave rise to the need of a clear taxonomy of the TDCDP, which was provided by a paper of Zhang et al. [20]. Importantly, Zhang divided the problem into 4 sub-problems [20], to which this paper will also adhere to[1]:

1. *Mixed delivery:* Trucks and drones both make a delivery. Furthermore, the drones are allowed to dock on the trucks. An example of this sub-problem is the previously described FSTSP [14].
2. *Drone delivery with truck-assisting:* Drones are the only vehicles that can make deliveries, while trucks have an assisting role in driving the drones to launching locations. Drones are allowed to dock on the trucks.
3. *Parallel delivery:* Trucks and drones independently make their own deliveries. The previously mentioned PDSTSP formulation [14] is an example of such a sub-problem.
4. *Truck delivery with drone-assisting:* Trucks are the only vehicles that can make deliveries, while drones have an assisting role in resupplying the truck with packages. Drones are allowed to dock on the trucks.

The scope of this literature review is limited to the modes where drones are allowed to dock on trucks, and are also allowed to make their own deliveries. This entails a focus on both *1. mixed delivery* and *2.*

[1] The order of the sub-problems was altered for clarity

truck delivery with drone-assisting, from hereon referred to as mode 1 and mode 2. When considering mode 1 and 2 delivery, several researchers have modified and extended the first formulation. In case the problem has a single truck and a single drone, it is most often called Traveling Salesman Problem with Drone (TSP-D). For example, Agatz et al. [13] later formulated the problem as an Integer Programming (IP) problem and proposed a route first-cluster second heuristics with dynamic programming. Carlsson and Song [21] used continuous approximation (CA) to formulate the problem, while Bouman et al. [22] formulated the problem as a MILP, and used dynamic programming to obtain exact solutions. Later, Bai et al. [23] considered the mode 2 TDCDP where drones can deliver multiple heterogeneous packages to customers on a real-world graph, while also allowing for launching locations that are not a customer location. This problem was named the Capacity-Constrained Heterogeneous Delivery Problem (CCHDP).

A major extension of the problem was introduced by Ferrandez et al. [24], who first formulated the TDCDP with multiple drones. They used the K-Means clustering algorithm to determine the launching locations of the drones and used a genetic algorithm to acquire solutions. Chang et al. [25] also took the the clustering approach for the truck multi-drone problem, but assumed only the drones can make deliveries. Several other papers used other heuristic approaches, for instance Murray and Raj [26], Luo et al. [27] and Tu et al. [28], who all considered that drone launching locations coincided with customer locations. Contrary to this approach, Young Jeong and Lee [29], Poikonen and Golden [30], Salama and Srinivas [31] and Mahmoudi and Eshghi [32] all considered the launch and retrieval (L&R) locations of the drones to be flexible and independent of customer locations. Leon-Blanco et al. [33] took a unique approach and used an agent-based formulation instead of an integer programming formulation, where agents represent points to be visited by vehicles, evolving within a grid according to set rules.

The problem was also extended to routing multiple trucks and multiple drones. In the formulations of Tong et al. [34] and Das et al. [35], each truck carries a single drone while serving customers, while Kitjacharoenchai et al. [36], Tamke and Buscher [37], Kloster et al. [38] and Kitjacharoenchai et al. [39] (this time with drones carrying multiple packages) all considered trucks carrying multiple drones.

To comprehensively represent the nuances present in the existing literature, an organized database was constructed of a sample of the available literature on the TDCDP and displayed by Table 2.1. The nuances that are present between the papers can be observed from the table, showing that no two papers are entirely alike across all criteria.

Some differences can be found in solving methods, where most of the research employs some kind of integer programming formulation, but some have other approaches such as CA or an agent-based formulation. Due to the NP-hard nature of the problem as discussed in Chapter 1, most of the papers use heuristics to efficiently solve the problem, while some use exact approaches such as commercial solvers.

Even more diversity can be found when considering the more specific characteristics of the model, such as whether or not heterogeneous drones are being used. This is done by Salama et al. [31] and Murray et al. [26], accommodating for the expansion of fleet with a variety of drones. The same classification can be made for the packages, and whether they are assumed to be heterogeneous or homogeneous. Bai et al. [23] assume heterogeneous packages since the drone can carry packages within its loading capacity. Other authors, like Murray et al [26] assume heterogeneous packages because of its influence on the endurance, and the capacity of the heterogeneous drones.

In terms of endurance, 4 different approaches can be identified that are used within the sample of existing literature:

1. *None:* An approach where drones are assumed to have infinite endurance, mainly used in papers purely intending to discover routing patterns independent of the range.
2. *Distance:* In this approach the drones have a maximum range they can cover from the truck and back.
3. *Time:* This approach is similar to the distance endurance, but now expressed in flying time. This usually gives a slightly more detailed representation of the drone's endurance than distance metric.
4. *Energy:* The energy approach is the most detailed one, where the drone has a certain energy level which starts draining based on for instance the flying time, weight of the drone, weight of the package, and distance covered.

Table 2.1: Classification of a sample of existing literature on the TDCDP mixed delivery (mode 1) and drone delivery with truck assisting (mode 2)

Authors and publication year	Problem name	Formulation	Solving method	# Trucks	# Drones per truck	Heterogeneous fleet	Heteregeneous packages	Drone multi-packages	Endurance model	Drone separation or queueing	Graph type	Non-customer launch	Launch ≠ retrieval	Delivery types
Bouman et al. (2017) [22]	TSP-D	MILP	Exact	1	1	No	No	No	None	None	Euclidean	No	Yes	T&D
Lin (2011)[10] [40]	VRPPDTW[2]	MIP	Exact, Heuristic	1	1	No	No	Yes	None	None	City map	No	Yes	T&D
Murray et al. (2015) [14]	FSTSP	MILP	Exact, Heuristic	1	1	No	No	No	Time	None	Euclidean	No	Yes	T&D,TO
Ha et al. (2018) [41]	TSP-D	MILP	Exact, Heuristic	1	1	No	No	No	Time	None	Euclidean	No	Yes	T&D
Agatz et al. (2015) [13]	TSP-D	IP	Exact, Heuristic	1	1	No	No	No	None	None	Euclidean	No	Yes	T&D
Carlsson et al. (2017) [21]	Horse Flies[1]	CA	Heuristic	1	1	No	No	No	None	None	Euclidean	Yes	Yes	DO
Bai et al. (2023) [23]	CCHDP[3]	MILP	Heuristic	1	1	No	Yes	Yes	Distance	None	City map	Yes	Yes	DO
Young Jeong et al. (2023) [29]	DRP-T	MILP	Exact, Heuristic	1	M	No	No	Yes	Time	None	City map	Yes	Yes	DO
Young Jeong et al. (2021) [42]	DRP-T	MILP	Exact	1	M	No	No	Yes	Time	Retrieval waiting	City map	Yes	Yes	DO
Poikonen et al. (2020) [30]	k-MVDRP[4]	IP	Exact, Heuristic	1	M	No	Yes	Yes	Energy	Assumed FCS[11]	Euclidean	Yes	Yes	T&D
Leon-Blanco et al. (2022) [33]	TmDTL[5]	Agent-based	ABM	1	M	No	No	Yes	Time	None	Euclidean	No	Yes	T&D
Salama et al. (2022) [31]	CTDRSP-FL[6]	MILP	Heuristic	1	M	Yes	No	No	Time	Coupled L&R	Euclidean	Yes	Yes	T&D,TO
Luo et al. (2022) [27]	MTSP-MD[7]	MILP	Exact, Heuristic	1	M	No	Yes	Yes	Energy	Assumed FCS[11]	Euclidean	No	Yes	T&D,TO
Mahmoudi et al. (2022) [32]	EM-TSPDs[8]	MILP	Exact, Heuristic	1	M	No	Yes	Yes	Energy	Retrieval waiting	Euclidean	Yes	Yes	DO,T&D,TO
Ferrandez et al. (2016) [24]	Clustered TSP[1]	MILP	Heuristic	1	M	No	No	No	None	None	Euclidean	Yes	No	T&D
Chang et al. (2018) [25]	Clustered TSP[1]	NLP	Heuristic	1	M	No	No	No	Distance	None	Euclidean	Yes	No	DO
Tu et al. (2018) [28]	TSP-mD	MILP	Heuristic	1	M	No	No	No	Distance	Retrieval waiting	Euclidean	No	Yes	T&D
Murray et al. (2020) [26]	mFSTSP	MILP	Exact, Heuristic	1	M	Yes	Yes	No	Energy	Coupled L&R[12]	City map	No	Yes	T&D,TO
Das et al. (2021) [35]	MOO TSP-mD[1]	MIP	Heuristic	M	1	No	No	No	Time	None	Euclidean	No	Yes	T&D
Tong et al. (2022) [34]	TSP-D	MINLP	Heuristic	M	1	No	No	No	Time	Retrieval waiting	Euclidean	No	Yes	T&D
Tamke et al. (2021) [37]	VRPD	MILP	Exact	M	M	No	No	No	Distance	None	Euclidean	No	Yes	T&D
Kloster et al. (2023) [38]	mTSP-DS[9]	MILP	Exact, Matheuristic	M	M	No	No	No	Energy	None	Euclidean	Yes	Yes	T&D
Kitjacharoenchai et al. (2019) [36]	mTSPD	MIP	Exact, Heuristic	M	M	No	No	No	None	None	City map	No	Yes	T&D
Kitjacharoenchai et al. (2020) [39]	2EVRPD	MIP	Exact, Heuristic	M	M	No	Yes	Yes	Distance	1 OP[13] per node	Euclidean	No	Yes	T&D

[1]No official name present in article
[2]Vehicle Routing Problem with Pickup and Delivery Time Windows
[3]Capacity-Constrained Heterogeneous Delivery Problem
[4]k-Multi-visit Drone Routing Problem
[5]Truck-multi-Drone Team Logistics Problem
[6]Collaborative Truck–Drone Routing and Scheduling Problem with Flexible Launch and Recovery Locations
[7]Multi-visit Traveling Salesman Problem with Multi-Drones
[8]Energy-constrained Traveling Salesman Problem - Drones
[9]multiple Traveling Salesman Problem with Drone Stations
[10]Although this formulation considers foot couriers instead of drones, the approach is very similar
[11]Flight Control System (FCS)
[12]Launch and Retrieval (L&R)
[13]Operation (OP)

When considering real-life operations, it is crucial to ensure separation between drones as to avoid potential collisions. While most papers focus on the routing and disregard the separation between the drones, a few do address this challenge. The most detailed variants can be found in papers by Salama and Srinivas [31] and Murray et al. [26], where the launch and retrieval operations are coupled and explicitly time-separated. Some other solutions are also possible, such as the model presented by Kitjaroenchai et al. [39]. In this specific model it is assumed that only a single operation is possible for each node to ensure separation between drones. Papers by Young Jeong and Lee [42], Mahmoudi and Eshghi [32], Tu et al. [28] and Tong et al. [34] consider a waiting time for the drone at the rendezvous location, but do not actively consider separation between drones.

Additional differentiation among the studies can be made on whether a city map is utilized to test the model, and if drones are permitted to launch from locations other than customer sites. For example, Mahmoudi and Eshghi [32] mentioned that ”If customer locations are considered the only possible nodes for drone launch and retrieval, some drone-eligible customers may not be within the flight range of a drone”. Salama and Srinivas [31] also demonstrated this exact importance, by showing that a substantial improvement in delivery efficiency can be achieved through the use of flexible launch and retrieval sites.

Furthermore, some papers like Ferrandez et al. [24] and Chang et al. [25] do not consider that the truck is moving while the drone is out making a delivery, i.e. there is no mismatch between the launch and retrieval locations of the drones that are launched from the truck. The main reason for not considering this aspect is because of the emphasis on the clustering aspect of the deliveries.

Lastly, the delivery types can be used to distinguish the papers. This is closely related to the delivery modes (recall that mode 1 signifies mixed delivery and mode 2 signifies drone delivery with truck-assisting). This specification shows the ’customer types’ of the model, where T&D indicates there exist nodes that can be services by either a truck or a drone. Some models also allow truck only (TO) nodes to exist, and others limit customers to only being able to be serviced by drones (DO). The model is not necessarily limited to one type of node. When multiple classifications are present in this column, it indicates that both types of customers are present in the model. When only DO deliveries are considered in the model, it suggests that it is a mode 2 delivery model. If either TO or T&D deliveries are present as well, then it is a mode 1 delivery model.

2.1.2. Collision avoidance and Airspace Simulation

Other studies consider path planning to avoid intrusions and collisions, which is a practical, real-life aspect of the routing process. These are absolutely crucial, since there are inherent uncertainties in the drone’s motion and environment. An example is the a paper by Wen [43], in which dynamic threats avoidance is modeled as a pursuit-evasion game. Primatesta et al. [44] take a two phase approach, generating a tentative static risk free route, and subsequently adjusting this route on-line with a dynamic risk map. A similar, dynamic, software-in-the-loop approach is taken in a study by Selvam et al. [45]. All of these studies use a simulated environment to verify the algorithms. Other papers, for instance one by Wan et al. [46], use a MILP formulation to develop a safe navigation environment, but do not create a simulated environment of the proposed system. Bahabry et al. [47] also formulate the collision free path planning problem as a MILP, but the results are in this case simulated using geographical data of Manhattan, New York City.

A comparison between multiple conflict resolution methods is performed by Ribeiro et al. [48], testing several common centralized and decentralized methods, e.g. the Modified Voltage Potential (MVP) [49], in the BlueSky air traffic simulator [19]. Research on organizing the drone airspace structure with fast-time simulations have been conducted by Sunil et al. [50], finding that vertical segmentation based on travel direction maximizes the airspace capacity. Sunil et al. [51] also put forward three-dimensional conflict count models, computing the instantaneous conflict counts in terms of traffic demand and other parameters. A useful taxonomy is also provided on intrinsic safety of an airspace, while also highlighting the difference between an intrusion (or loss of separation) and a conflict. When considering real world applicability, these two concepts are aspects that must be considered.

2.1.3. Dynamic Traveling Salesman Problem Models

Although none of the research described in Subsection 2.1.1 contains dynamic conditions and uncertainties, there are several studies that do consider such an environment. In Chapter 1, the interest in algorithms that do not assume complete predictability was highlighted. To find papers that account for dynamic conditions rather than entirely deterministic inputs, the regular Traveling Salesman Problem (TSP) is examined. For instance, one paper addresses a scenario where customer request locations may change over time, considered by e.g. Zhang et al. [52]. A strategy is proposed where the decisions of the salesman are continuously revisited after a delivery is completed. This approach also ensures that traffic conditions are taken into account, since the weights between the customers can be changed every time a customer is serviced. The inclusion of weights that can be changed throughout the analysis is referred to as weight shifting. A deep reinforcement learning algorithm is proposed to tackle this problem. Similar weight shifting conditions are considered by Mavrovouniotis et al. [53], where a memetic Ant Colony Optimization (ACO) algorithm is proposed to efficiently tackle the dynamic TSP. Stochastic elements are further introduced into the problem by considering uncertain package release dates of packages at the depot, formulated by Archetti et al. [54]. Packages must be picked up at the depot before they can be delivered, and the time when a package can be picked up is not known beforehand.

2.2. Conclusions of Literature Analysis

A key observation that can be made based on Section 2.1 is that none of the existing papers consider a dynamic environment, in which real-life uncertainties are included. The algorithms in literature assume complete predictability of the conditions during the delivery process, such as delivery times and travel times. The existing TDCDP literature solely focuses on the routing aspect of the trucks and drones under deterministic, known inputs and disregards practical feasibility and applicability of such solutions. For instance, what happens when the truck gets caught in a traffic jam, or when one of the deliveries takes a longer time than expected? Since all optimization algorithms are pre-planned solutions, no deviation from this route is not accounted for.

The conclusion that can be drawn from the analysis of TDCDP literature is that the models that solve the problem have become very powerful. Even though it is a NP-complete problem, as discussed in Chapter 1, heuristic algorithms are capable of finding near optimal solutions in every distinct variation of problem. However, because there is no feedback on status of the network (i.e. position of the truck, updates on deliveries, situation of road traffic, etc.), the difference between a real-world execution of the optimized solution may perform worse than calculated beforehand. This is especially feasible when uncertainty aspects are introduced to the optimization problem. The existing literature thus provides a mathematically optimized, yet relatively low-fidelity solution due to the absence of practical feasibility and implementability. The only instance where uncertainties are introduced, and thus where the algorithm also dynamically accounts for these cases is for the Traveling Salesman Problem without any drones.

Furthermore, it becomes clear that although there have been numerous studies into airspace and air traffic simulation, no simulations have been performed of a truck-drone delivery network, and there have also been no previous studies on collision avoidance considering this type of airspace occupation. Numerous studies have developed mathematical models for the routing of trucks and drones. Similarly, several studies have been published on collision avoidance and airspace structure, also considering simulations of airspace. However, no papers are present in literature that consider a simulated variant of the TDCDP, ensuring a both efficient and safe airspace.

3
Research Outline

In this chapter, the approach of filling the research gap that was identified in Chapter 2 is described. The research objective and questions are first stated in Section 3.1. The requirements and selection of the existing paper from literature are discussed in Section 3.2. The hypotheses to the research questions are formulated in Section 3.3.

3.1. Research Objective and Questions

From the literature review, it became clear that there is is an abundance of mathematical models that aim to efficiently solve the combined delivery routing problem of a truck and drone with deterministic. Furthermore, the importance of dynamic routing under realistic conditions and simulation of airspace was highlighted. From the observations that were made in that chapter, the objective of this research is to address that specific gap. This gap is visualized in Figure 3.1. More specifically, existing truck drone routing literature will be used to compare a novel, dynamic model in under disruptions and uncertainties. The comparison will be made in a simulated environment designed to represent real-world conditions as closely as possible.

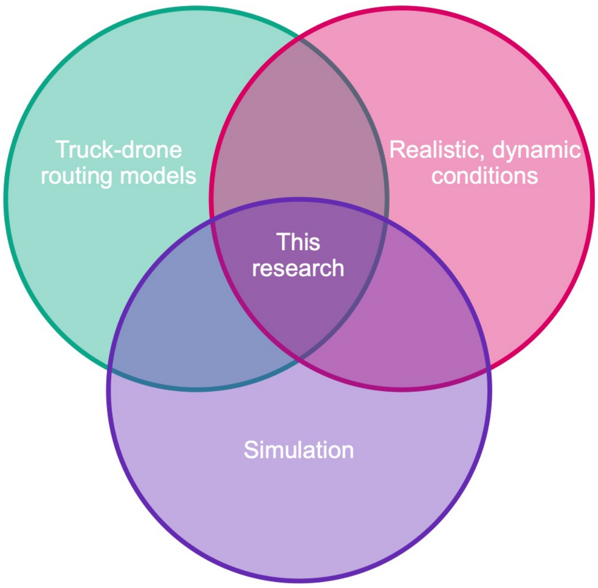


Figure 3.1: Visual representation of the research gap as identified in the literature review

The objective that is tied to this visualization is formulated in the Research Objective box.

Research Objective

The research objective is to construct a novel and dynamic model for the Traveling Salesman Problem with multiple Drones (TSP-mD) and to compare its performance in terms of efficiency (e.g. completion time, drone waiting time, delays due to traffic or delivery duration deviation) to an existing model in a dynamic environment. This will be achieved by running a vehicle simulation in the Bluesky environment.

In order to successfully achieve this objective, a set of Research Questions is developed. The main research question is presented above.

Main Research Question

What are the efficiency gains of introducing a dynamic TSP-mD model when compared to an existing TSP-mD model from literature in a dynamic and realistic environment?

To answer this question is a step by step manner, 3 sub research questions have also been established. The first two of these assist with setting up the comparison of the two models, while the last sub question focuses on breaking down the 'efficiency gains' term in the main research question. This means that the first two questions are less directly related to answering the research question, but offer a more comprehensive breakdown of the project plan that is presented in Chapter 4. The first one of these assisting sub questions addresses the selection of a model from literature, and also touch upon the assumptions and adaptations that are required. It is shown below.

Sub Research Question 1

Model selection, specifications and reformulation:

(a) What criteria and selection process can be employed to identify the most suitable existing TDCDP model from literature for adaptation?
(b) What procedure can be employed for effectively simulate the existing model in a Air Traffic Management (ATM) simulation?

The second sub research question is also an assisting question, and aims to put the performance of the model into perspective by introducing a novel, real-time algorithm. It is:

Sub Research Question 2

How can a novel real-time algorithm be designed and implemented to solve the truck and drone cooperative delivery problem (TDCDP) mixed delivery?

The last sub research question dives into measuring the performance of a TDCDP model, required to offer an even ground for comparison. It is formulated as follows:

Sub Research Question 3

a) How does the difference in terms of mission completion time (makespan) between the two models compare when subjecting it to different levels of uncertainty, and is there a benefit from using the dynamic algorithm in more uncertain conditions?
b) How do the the models compare in terms of the distance that the drones and truck traverse?
c) How is the waiting time at the pickup location of either the truck or drone affected by increasing the uncertainty?

3.2. Model Selection

A representative paper is to be selected in order to address this set of research questions. When considering the literature review as explained in Chapter 2, a few notable aspects can be highlighted:

1. First of all, it had to be decided whether a mode 1 or a mode 2 delivery network is to be considered. Of the two options, mode 1 is considered the most general delivery network in urban areas, which is the focus of the research;
2. Furthermore, because of the complexity of the problem, it is desired the problem is formulated with only 1 truck;
3. With the concession of having only 1 truck, it is desired to have multiple drones on this truck, to address crucial sequencing and timing aspects of such a delivery network;
4. Since this research will focus on realistic conditions, it is desired that the model has been tested on a real city graph;
5. These realistic conditions also imply the use of a realistic endurance model. Therefore, only papers that consider energy endurance are desired, since this is the most complex variant.

Using Table 2.1, it can be concluded only one paper meets all these requirements. The name of this article is "*The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones*" (2020), by Chase C. Murray and Ritwik Raj [26]. This paper will be used as a representative article from literature during this research.

3.3. Hypotheses

When considering the existing TDCDP, they are often complex and able to solve the problem near optimally. However, as previously discussed, none of the papers include put forward a robust model that is able to respond to disruption and uncertainties. Therefore, the models can actually only handle deterministic inputs without any deviations from these values, and cannot respond to en-route conditions. A simulation will be used to validate the model's effectiveness in this conditions. Unexpected conditions could deteriorate the performance of the model, for example delayed delivery times or wind.

It is expected that the model's performance is significantly harmed by these conditions. This could be caused by the need of having to adapt to these altered, uncertain conditions. Even though the existing models are very complex, they are a simple representation of the conditions one would face in real life. The dissimilarity between the two cases—realistic versus simplistic conditions—is expected to significantly impact performance to such an extent that the complexity of the model might be excessive. Therefore, it is also expected that a much simpler, but dynamic model can get a more or less equal performance when being benchmarked against an existing model in terms of mission completion time. Furthermore, it is expected that the distance traveled by the vehicles will increase slightly, but that the waiting time will significantly be reduced at the same time.

4 Project Plan

This chapter concerns the project plan that will be employed throughout the thesis. This plan is constructed to meet the required deadlines in time, and to timely answer the research questions and fulfill the research objective that have been stated in Chapter 3. The methodology for answering the research questions and the sub question is first discussed in Section 4.1, along with the expected results that should follow from this methodology in Section 4.2. The planning for the project that is be adhered to is outlined in Section 4.3. Lastly, the execution of the planning in retrospect is discussed in Section 4.4.

4.1. Methodology

The backbone of the methodology is the Bluesky simulation environment. The existing Truck Drone Collaborative Delivery Problem (TDCDP) model will be tested and simulated in this environment. The performance of the model will be derived from the simulation results.

While Bluesky is an advanced air traffic simulation tool, it does not yet support the TDCDP. However, since Bluesky is a modular tool, plugins can be used to extend its capabilities. Essentially, multiple plugins can be created to support the TDCDP simulation, thereby extending its use cases to also include the TDCDP scenario.

As outlined in Section 3.2, the article that will be used to simulate the existing model is the 2020 paper by Murray and Raj [26]. Conveniently, their model is available on Github[1]. Solutions of existing problems are available as well on the same Github repository. These solutions will serve as the basis of comparison in this research. They are performed in Seattle (a large city) and Buffalo (a smaller city), and solutions to the problems are provided as .CSV files. Several different other parameters are taken into account as well, such as number of drones, specifications of the drones, and number of customers.

A crucial part of the project is to correctly simulate the solutions that are provided by this paper. For this purpose, a scenario converter will be developed to convert these .CSV solution files to .SCN BlueSky scenario files. The aim is to provide a fully automated converted, such that the entire process is done by the converter, and such that these solutions can be directly simulated. To achieve this, the custom plugins will be used to enable the functionalities of delivery, and launching and retrieving drones.

A novel online algorithm will be developed to solve the same instances as the existing model by Murray and Raj [26]. The inputs (i.e. customer locations, number of drones, drone specs, etc.) will be taken from the solutions of the existing article such there is no discrepancy between these. The results of both models will later be compared to each other, to enable measuring the performance in terms of the self-established metrics. Possible takeaways from this comparison will also be used to further improve the novel model. After completing the development of the two algorithms, the comparison is conducted using metrics that have been defined prior to the comparison. These metrics assess the safety (e.g. drone separation) and efficiency (e.g. flight time and computational intensity) of the algorithms in a simulated real-world setting within the ATM Bluesky environment. Following this evaluation, the two models are compared to highlight their respective strengths and limitations, providing insights into their practical applicability and overall performance.

[1] https://github.com/optimatorlab/mFSTSP

A visual representation of the entire pipeline described in this section can be found in Figure 4.1. The instances are already generated by the existing paper, and are being fed to the novel algorithm, For the existing algorithm, the back module is used to convert the solution file to a Bluesky scenario (.scn) file. This is then fed to the Bluesky simulation itself.

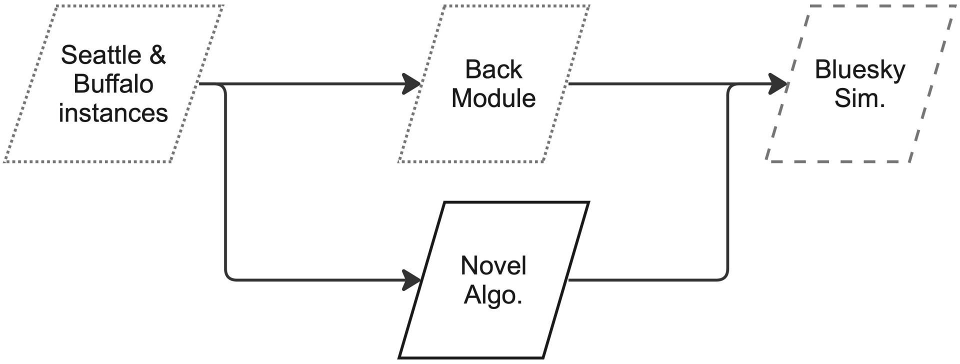


Figure 4.1: Workflow pipeline used to test the algorithms in Bluesky

4.2. Expected Results

The methodology explained in Section 4.1 is expected to deliver a simulation of the heuristic model described by Murray et al. [26] in Bluesky, based on the solutions provided by this model. Furthermore, the performance of this model compared to the novel algorithm is also an expected result, generated by a complete, simulated head-to-head comparison. The comparisons will be made under different conditions, for instance the benchmark where no unexpected conditions occur. To introduce dynamic conditions, traffic jams will be introduced as well as delays in delivery, or potentially deliveries that are completed quicker than expected. The results of these distinct scenarios are expected to serve as a basis to perform a quantitative analysis on. Finally, it is expected one of the two models performs better than the other such that the hypothesis stated in Chapter 3 can be accepted or rejected.

4.3. Planning

To ensure timely delivery of all deliverables, and to ensure fulfillment of the research objective and answering of research questions, the planning was divided into work packages. These work packages can be found in Table 4.1. Work package 0 is about getting started with the thesis, which has already been completed. The main deliverable for the first 'official' work package is this document, serving as the research proposal. The activities for this work package also entailed an extensive literature review, which resulted in Chapter 2. Furthermore, this planning was drafted and preliminary testing was performed.

The second work packages revolves around setting up the Bluesky workflow and pipeline. This consists of extending the vanilla version of Bluesky with custom plugins as described in Section 4.1. Furthermore, the front and back modules are created as part of this work package, as also described in Section 4.1 and depicted in Figure 4.1.

Work package 3 dives into the existing heuristic model's solution. In this workpackage, conversion from the .csv files to .scn files will be setup. This allows simulation in the Bluesky environment.

To compare the engine to another model, work package 4 constructs a novel model that can solve the problem. In this work package, the specifications of the algorithm are identified, and the algorithm is implemented and tested.

Work packages 5 is centered around comparing the 2 models, and identifying and incorporating potential modifications that can improve the existing model even more. Afterwards, the performance of the two models is again compared, if any modifications have been made.

The last work package, work package 6 is focused on writing the thesis report and preparing the thesis defense to successfully wrap-up the thesis.

A Gantt chart is also detailed of this project and can be found in Figure 4.2, containing all work packages that have been described in Table 4.1. It should be noted that while the sub-work packages have their own tasks, these are not included in the Gantt chart to maintain clarity.

4.4. Reflection on Planning

This section is included as a to reflect on the planning that has been drafted at an earlier stage of the thesis. The planning was last revisited in the end of May 2024, approximately a week before the midterm. Overall, the planning that was drafted was closely following, resulting in a tight but well-controlled trajectory. Although the intermediate planning was sometimes refined slightly, the deliverable dates were barely changed. the initial green light meeting was approximated to take place on the 4th of November 2024, which is now scheduled for the 5th of November 2024. This indicates that in general the planning was successfully completed.

Table 4.1: Work package breakdown

WP ID	Work Package Name	Description
WP 0	**Thesis startup**	Start thesis, organize documents, etc.
WP 1	**Literature Review and Research Proposal**	Reviewing existing literature and establishing the thesis Research Proposal
WP 1.1	Read articles and article selection	Reading relevant articles on the TDCDP, simulation of air traffic, etc. and selecting relevant articles
WP 1.2	Draft Research proposal	Creating this document and presenting it
WP 1.3	Draft the planning	Creating the planning outline over the entire thesis duration
WP 1.4	Start testing	Start performing small scale Bluesky tests to get familiar with the functionalities.
WP 2	**Setting up Bluesky workflow**	Establishing the front and back module
WP 2.1	Develop problem generating module	Develop the problem generating module (front module), generating random instances to be tested with
WP 2.2	Set up scenario converter	Develop the engine solution to Bluesky scenario converter (back module), generating scenario files based on the solution of the engine
WP 3	**Simulate existing model solutions**	Simulate the solutions of the existing model in Bluesky
WP 3.1	Retrieve and process osmnx graph	Get the graph in osmnx and prepare it for usage
WP 3.2	Setup truck routing	Set up correct routing of the truck on the graph
WP 3.3	Setup drone routing	Setup euclidean routing of the drones
WP 3.4	Setup translation to Bluesky scn file	Convert all routes to Bluesky scenario
WP 3.5	Verify Bluesky compatibility	Check whether scenario is correctly functioning in Bluesky
WP 3.6	Set up KPI measuring	Setup extraction of KPIs during the simulation
WP 4	**Design and implement novel algorithm**	Designing and implementation of the novel algorithm to be compared with heuristic model
WP 4.1	Determine algorithm specifications	Determine what the model's algorithms rules are going to be and develop pseudo code
WP 4.2	Implement algorithm	Code and implement the algorithm
WP 4.3	Testing the code implementation	Ensure proper functioning of the code, bug fixing where necessary
WP 5	**Model comparison and improvement**	Head-to-head comparison of the two models in terms of the same predefined metrics
WP 5.1	Set up batch simulation testing	Determining format of simulation testing, e.g. Monte Carlo simulations
WP 5.2	Process results of testing	Collecting and processing results that are obtained from the batch simulation tests
WP 5.3	Performance comparison and potential improvements	Comparison based on the results and optional reiteration of the models
WP 6	**Thesis wrap-up**	Final stage of thesis, wrap-up consisting of report writing and thesis preparation
WP 6.1	Write thesis report	Writing of thesis report document
WP 6.2	Prepare green light review	Preparation of the green light meeting
WP 6.3	Prepare thesis defense	Prepare defense presentation

Figure 4.2: Gantt chart of the thesis planning. Sub-tasks are omitted for clarity

Part III

Supplementary Work

5 Code Structure

To manage the diverse coding topics covered in the project, individual repositories are established to ensure a clear separation of code. Given this intricate structure, this chapter is dedicated to discussing the code organization and functionality. Firstly, the architecture is visualized in Section 5.1. Afterwards, the important factors of individual repositories are highlighted. These explanations start with the graph generation in Section 5.2 followed up by the simulation of the reference paper in Section 5.3.1. Then, the repositories that are used to route the truck are explained in Section 5.4. The end goal, which is the simulation in Bluesky, is explained in Section 5.5. Lastly, a collection of all links to the repositories that were used is given in Section 5.6.

5.1. Code Architecture

This section regards the structure of the project's code. Since several repositories were used for the study, an outline of the collaboration between the different repositories is given in Figure 5.1. Important functions are also outlined in the figure.

5.2. Graph Generation

The graph data in this paper is obtained from OpenStreetMap [55], and the `OSMNx` library is used to access this data [56]. While a significant amount of data is contained within the graph, some modifications and additions require to be made to the graph. To ensure that this happens in a consistent manner and that the same graph data is used consistently throughout the paper, a separate repository was developed, called `GraphGen`.

First of all, the graph geometry that is provided by `OSMNx` is highly detailed. When translating this data to Bluesky, the number of waypoints becomes excessive and unmanageable. Therefore, the graph is simplified before usage. By visually inspecting the graph in QGIS, a setting is obtained which ensures a detailed geometry of edges in the graph, but also ensures that the amount of waypoints that are required is limited.

Next to the simplification, some missing data must also be inserted. Some edges of the `OSMNx` graph do not have a speed attribute, which is required for the truck routing in Bluesky. The procedure of filling these gaps is done as follows:

- The edge data that is present is processed. For each 'highway type', the most common speed is stored.
- The edges that have missing speed limits are now processed by highway type. Each edge inherits the most common speed limit for its type.
- Edges that still do not have a speed limit after the previous steps are given the most common speed in the graph in general.

With all speed limits defined for the edges, the traversal time can now be calculated by dividing the length of each edge by its respective speed limit.

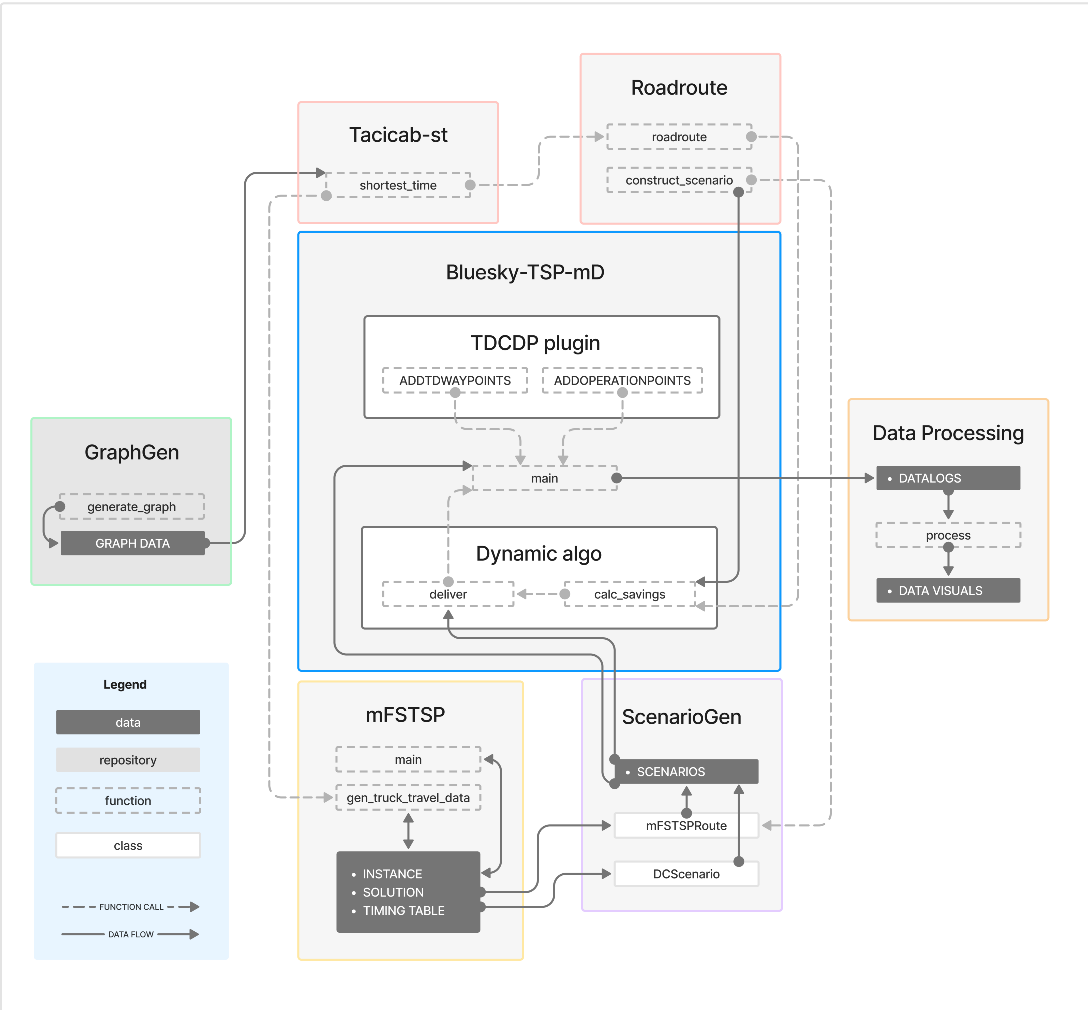


Figure 5.1: Code Structure of the Project

By generating the graphs that are used in the experiments with this graph generation module, consistency of the data and routing is ensured.

5.3. Simulation of Reference Paper

To capture the solutions of the reference paper in a simulation, a careful procedure is followed to ensure that it is accurately reflect it. Firstly, this consists of ensuring that the inputs of the model are consistent with the inputs that are provided to the dynamic algorithm. The method that is used for this is outlined in Subsection 5.3.1. Once it has been assured that the inputs are consistent, the algorithm is called, resulting in a set of solution files. The routine that is consequently followed to translate the solutions to a Bluesky .scn file is explained in Subsection 5.3.2.

5.3.1. Reference Paper and its Inputs

The inputs for the reference paper consist of a timing table in which the time required to drive from any customer A to any customer B is given. This time required is calculated by the Taxicab-ST module, which is explained in more detail in Subsection 5.4.1. The module calculates the quickest route and returns the corresponding value in seconds. This value is then stored in a .csv file (called *tbl_truck_travel_data_ <problemname>*). Furthermore, euclidean distances between the customer combinations are also required. This is used by the reference paper to determined if a drone trip is within its range. These euclidean distances are calculated with the geopandas `geodesic` function. These values are also stored in the same .csv file. With this data present, the algorithm can be called, which generates the corresponding solutions in a different .csv file. This solution .csv file follows the naming convention *tbl_solutions_<drone_type>_ <drone_count>_Heuristic*, where the drone type considered is always type 101. This corresponds with the *high speed, low range* drone. The .csv file contains some metadata, after which the assignments of the truck and drones are given. The generation of the truck timing table, as well as the solution files is performed in the `mFSTSP` repository. This repository also contains all customer instances that are used for the experiments. For each customer count (10, 25, 50, and 100), there are a total of 10 scenarios present. A total of 4 distinct uncertainty settings are used, as is explained in Subsection 5.3.3. Accounting for 4 different drone counts, a total of 640 scenario files of the reference paper are generated.

5.3.2. Translation to Bluesky Scenario

After having obtained the solution .csv files, the assignments that are obtained can be translated to a scenario file. The module responsible for for this conversion is called `ScenarioGen`. Each row of the assignments is analyzed by identifying its contents. Firstly, the truck route is extracted. This is done in the function `construct_truck_route`. The truck tour followed from every customer A to B, for which the route between is generated using the `Taxicab-ST` module, which is further detailed in Subsection 5.4.1. The resulting route is then converted to a list of coordinates, along with speed limit for each coordinate and the estimated time of arrival (ETA). This data later serves as input for the `construct_scenario` function of the `Roadroute` module. After the truck waypoints are identified and added to the scenario file, the truck deliveries and drone sorties are extracted in `get_deliveries` and `get_sorties` respectively. This data is then converted to scenario format in `delivery_scen` and `sortie_scen`. By chaining all the individual commands to a single scenario file, the completion of the scenario is completed and ready for simulation in Bluesky. Note that these scenario files make use of plugins that are required for the simulation, which are described in Section 5.5.

5.3.3. Incorporation of Uncertainties

The uncertainties that are introduced in the simulation must be consistent for both models, i.e. for the reference paper and for the dynamic algorithm. Therefore, these must be generated in a central location, where they can be applied to both models. The most convenient way to incorporate them into the simulation was to include them in the scenario file, by directly commanding the truck stop operations, altered customer service times, and the modified drone speeds. These uncertainties are then incorporated not only to the scenario file of the reference paper, but also in the scenario of the dynamic algorithm. In total, 4 distinct settings are used to vary the uncertainty. They are:

- 0: Base uncertainty level, without any uncertainty.
- 1: Low uncertainty, for uncertainties that should be very common.

- 2: Medium uncertainty, for uncertainties that should occur occasionally.
- 3: High uncertainty, with uncertainties that shouldn't occur often.

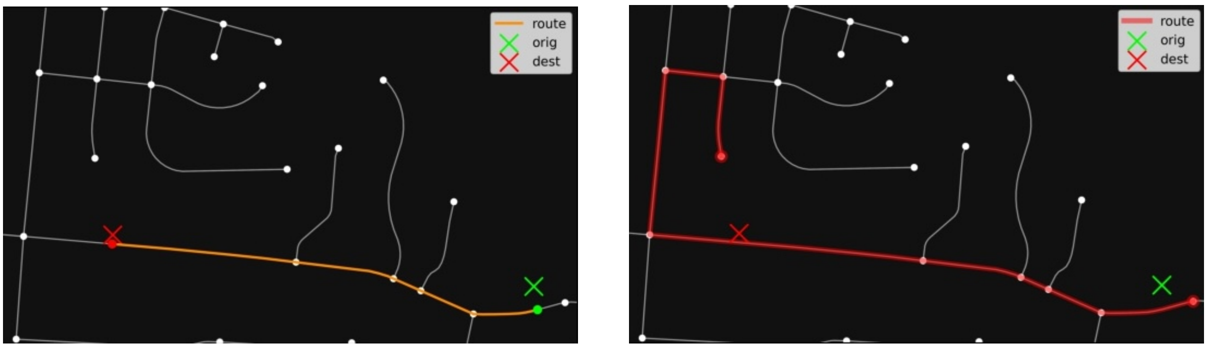
The implementation of the uncertainties is relatively straightforward. There are 3 types of uncertainties that were implemented: truck stopping at a certain frequency, modified drone speed, and modified service time. Firstly, the alteration of drone speeds is accomplished simply by setting a modified speed for the drone's waypoints. Furthermore, the modified service time is realized by modifying the stand still time for delivery, which is a Bluesky extension as explained in Subsection 5.5.1. Lastly, the truck stopping required its own operation type. This exact functioning is detailed in the same subsection.

5.4. Truck Routing Modules

Truck routing is a separate module in the coding architecture, since its functioning is complex. The customer locations are provided by the instances in the `mFSTSP` repository are crucially not located directly on the nodes on the graph, and in fact are also not necessarily positioned along an edge. Also, the dynamic algorithm might be rerouted at any point in time, which results in its position also being at any position along an edge. Therefore, advanced routing is needed to determine the quickest route between two points. This routing is divided into two parts: firstly, the calculation of the quickest route from point A to point B is explained in Subsection 5.4.1. Then, the conversion to coordinates is explained in Subsection 5.4.2.

5.4.1. Taxicab-ST Shortest Time Routing

To calculate the shortest path between two points, an out of the box method cannot be used. These methods calculate the shortest path between two nodes, and do not account for the fact that the locations might not be situated on a node. In this study, the shortest path is therefore constructed by the `Taxicab-ST` module. The function `shortest_time` first calculates the nearest edge of the starting location, and also selects the point on the edge that is nearest to that edge. The same procedure is then followed for the destination location. Then, the nearest node that is connected by the edge is selected for both the origin and destination. These serve as the start and end point of the core of the route, which is still the networkx function `shortest_path`. This function requires the start and end node of the route to be parsed, and returns a sequence of nodes that construct the core of the shortest path. The Dijkstra algorithm is used to obtain this path. Some example use cases of the `Taxicab-ST` module are given in Figures 5.2, 5.3, and 5.4, all taken from original `Taxicab` repository[1].



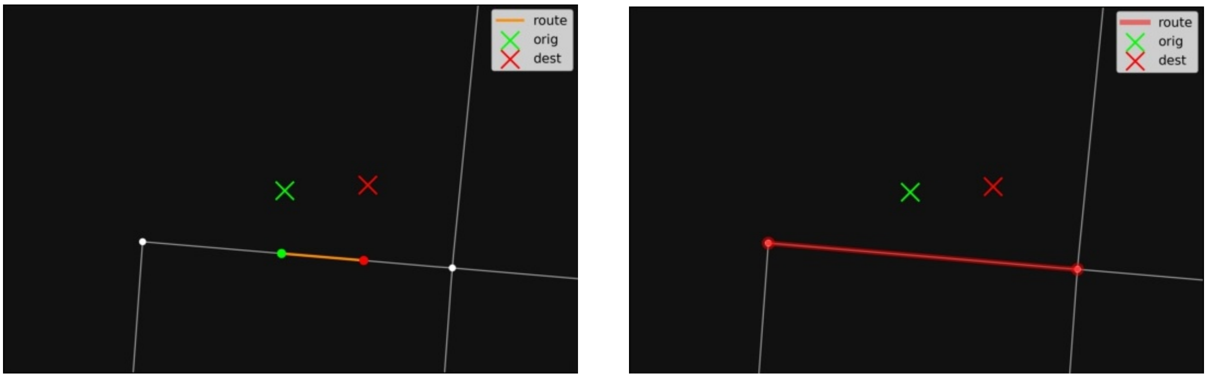
(a) Route Generated by `Taxicab-ST` **(b)** Route Generated by `OSMNx`

Figure 5.2: Difference between the `OSMNx` Route and `Taxicab-ST` Route Between Distant Points

The first example, shown in Figure 5.2, demonstrates the use where a route is desired between two points. A significant amount of unnecessary route is removed, and overall the route is much more realistic.

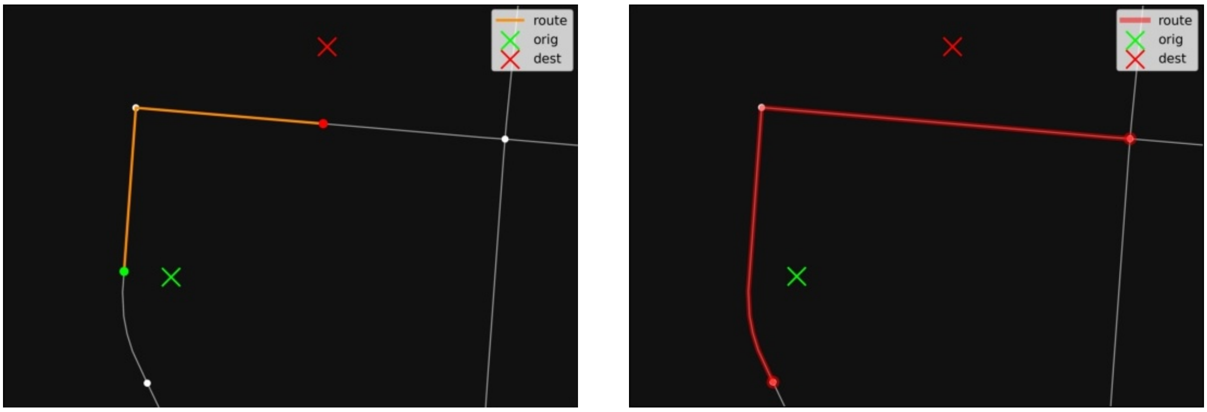
The second example considers a route between two points on a single edge. Significantly more accuracy is required in this case, as the `OSMNx` routing module returns a route of more than double the size of the `Taxicab-ST` module. This example is shown in Figure 5.3

[1] `https://github.com/nathanrooy/taxicab`



(a) Route Generated by `Taxicab-ST`   **(b)** Route Generated by `OSMNx`

Figure 5.3: Difference between the `OSMNx` Route and `Taxicab-ST` Route on a Single Edge



(a) Route Generated by `Taxicab-ST`   **(b)** Route Generated by `OSMNx`

Figure 5.4: Difference between the `OSMNx` Route and `Taxicab-ST` Route Between Nearby Points

The last example is illustrated by Figure 5.4. This is an example of a shorter route where the benefits of the `Taxicab-ST` module are also evident, since 2 edges are partly selected instead of 2 complete edges.

The algorithm also return the total time required for the route to be completed, which is calculated from the sum of all individual edge parts as well as the beginning and ending line pieces. These follow directly from the edge properties explained in Section 5.2. Note that every coordinate requires an estimation of the time that it takes to be completed, hence it might be the case that there are several estimated times for the traversal of a single edge.

5.4.2. Conversion from Taxicab to Coordinates

Once the nodes that make up the route, as well as the beginning and ending line pieces have been obtained, they can be translated to a sequence of coordinates, which is done within the `Roadroute` repository. With the `roadroute` function, the sequence is constructed mainly from the data of the edges that are part of the route, while the beginning and ending part are appended to it. Furthermore, the edge travel speed limits are extracted from the edges, which is another element that is being returned by this function.

Then, the `construct_scenario` can be called to translate these coordinates into Bluesky TrafScript addwaypoint commands. For this, the previously generated speed limits are also required. As explained in Subsection 5.4, this function is directly used to generate the truck route for the scenario generation.

5.5. Bluesky Simulation and Plugins

In this section, the modifications that are made in the Bluesky repository itself (`Bluesky-TSP-mD`) are explained. Since operations are not part of vanilla Bluesky, this is added to the simulation with a plugin. The procedure to do so is explained in Subsection 5.5.1.

5.5.1. Operation Handling

Bluesky's primary aim is not to simulate a truck and drone delivery network. Therefore, custom Bluesky commands must be designed that facilitate convenient simulation of the delivery network. Bluesky's modular setup allows to address the specific challenges posed by drone-based delivery systems in urban environments by adding additional functionalities. These functionalities consist of the operations that are possible within the delivery network. Every operation (except the truck stops), as well as the completion of the mission is logged to a separate log file. This allows for traceable performance management of the operations, which consist of the following:

1. Delivery: A delivery consists of a vehicle stopping at a customer location and remaining there for a designated amount of time;
2. Sortie: A sortie is the process of stopping at a defined point, and launching a drone at standstill from that location. This process also takes a certain amount of time, during which the truck must remain stationary;
3. Rendezvous: The rendezvous functionality represents the retrieval of a drone, performed by the truck and its driver. It requires the truck and drone to be stationary for a given amount of time. Both vehicles must remain at the same location during the entire duration
4. Stop: An equal operation as the delivery type, but if this type is selected, the operation will not be logged. This operation type is required for the truck stopping uncertainty as described in Subsection 5.3.3.

The custom TrafScript commands `ADDTDWAYPOINTS` and `ADDOPERATIONPOINTS` were developed to enable a solution to perform either 4 of these operations. The former is an extension of the the already existing TrafScript command `ADDWAYPOINTS`, but allows the waypoints to be modified such that it contains an operation. These operations can be added with the latter command. Both of these functions are an extension of the existing `Route` implementation, which is replaced with `TDRoute` which includes these two new functionalities.

Further modifications of the existing Bluesky functionalities include changes to the autopilot as well as the activewaypoint implementations. These modifications can be found in the scripts called `TDAutopilot` and `TDActWp` respectively. These plugins are also required to simulate the truck and drone deliveries and ensure proper functioning of the previously mentioned operation handling and routing of the truck and drones.

5.6. List of Links to Repositories

This section contains a list of all links to the repositories that were used. They are the following:

- `Bluesky-TSP-mD`: `https://github.com/ravenvanewijk/bluesky-TSP-mD`
- `Roadroute`: `https://github.com/ravenvanewijk/roadroute-lib`
- `Taxicab-ST`: `https://github.com/ravenvanewijk/taxicab-st`
- `GraphGen`: `https://github.com/ravenvanewijk/mFSTSP-GraphGen`
- `ScenarioGen`: `https://github.com/ravenvanewijk/mFSTSP-ScenarioGen`
- `mFSTSP`: `https://github.com/ravenvanewijk/mFSTSP`

6
Additional Results

6.1. Mission Completion Time

In the paper, an Analysis of Covariance (ANCOVA) is performed on the Mission Completion Time, to investigate the impact of uncertainty, dronecount, and customercount on the mission makespan. These results are elaborated upon in this section, by investigating the individual p-values of different setups of number of customers and number of drones.

The distribution of the samples of ΔMS are observed to be normally distributed in the paper by using the Shapiro-Wilk test. Hence, the double sided T-tail test can be used to investigate the relative performance of the DA w.r.t. the RM when considering a change in uncertainty. For this purpose, the null hypothesis is formulated to state that there is no difference in the percentage differences in ΔMS between two uncertainty levels. The percentage differences of groups 1, 2, and 3 are compared to the control group without uncertainty, i.e. uncertainty level 0. Since the tests for every level consist of 16 post-hoc tests, the significance level is reduced to $\alpha = \frac{0.05}{16} = 0.003125$. This is due to the Bonferroni correction, as a measure for the multiple comparisons. The p-values results are shown in Tables 6.1, 6.2, and 6.3 for uncertainty levels 1, 2, and 3 respectively. P-values greater than 0.003125 are colored red, while values lower than 0.003125 are colored green.

Table 6.1: P-Values of Double Sided T-tail Test for Δ_{perc} of Uncertainty Level 1 Against Level 0

Customer Count	1 Drone	2 Drones	3 Drones	4 Drones
10	0.342	0.768	0.756	0.934
25	0.747	0.899	0.839	0.327
50	0.438	0.815	0.782	0.408
100	0.196	0.845	0.528	0.799

As expected from the distributions of Δ_{perc}, there exists no statistical difference between uncertainty level 1 and the control uncertainty level 0. This applies to all distinct combinations of drone count and customer count. It can even be observed that the p-value of 1 drone and 100 customers comes closest to being significant, where the RM improves when compared to the DA.

Table 6.2: P-Values of Double Sided T-tail Test for Δ_{perc} of Uncertainty Level 2 Against Level 0

Customer Count	1 Drone	2 Drones	3 Drones	4 Drones
10	0.297	0.986	0.852	0.723
25	0.537	0.367	0.281	0.649
50	0.566	0.068	0.579	0.132
100	0.945	0.916	0.969	0.287

As the uncertainty increases to level 2, there still exist no setup with statistical significance between the two levels. Thus, the result do not confirm nor deny a statistical difference between the Δ_{perc}. This can either be attributed to the fact that the sample size is too low to prove any significance, or that there is no statistical difference between the two levels.

Table 6.3: P-Values of Double Sided T-tail Test for Δ_{perc} of Uncertainty Level 3 Against Level 0

Customer Count	1 Drone	2 Drones	3 Drones	4 Drones
10	0.021	0.496	0.007	0.073
25	0.017	0.058	0.017	0.101
50	0.008	0.000	0.005	0.009
100	0.018	0.014	0.002	0.001

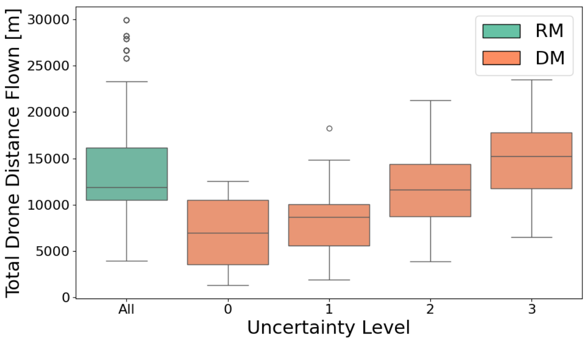
In uncertainty level 3, however, the null hypothesis can be rejected for 3 out of the 16 cases. This entails that the relative improvement of the DA with respect to the RM is significant in some measured setups. Crucially, 2 setups with 50 or 100 drones show a p-value lower than 0.05, indicating that there is a significant difference for these setups. This is likely due to the fact that uncertainties can more easily propagate in these scenarios, since more customers need to be served which might cause the uncertainties to pile up and the solution quality of the RM to degrade. While this is an interesting observation, it must be recalled that the DA still has a higher median makespan than the RM in nearly all conditions. This is especially true for the setups with 50 or 100 customers, where the RM always outperforms the DA.

A noteworthy observation is that the differences become statistically significant only in the highest uncertainty level, with no gradual progression but rather a sudden shift. One possible explanation for this observation is that the parameter increments for each level may be too large. Possibly, there is a threshold value for one or more of the uncertainty parameters beyond which the RM solutions quickly deteriorate, which could be a value between levels 2 and 3.

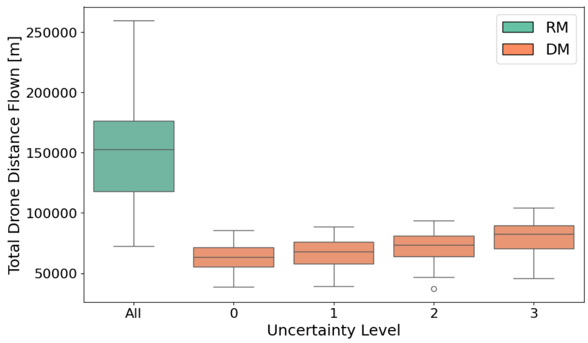
6.2. Distance Traveled

Next to the analysis of the impact of the model on every one of the customer counts in one single analysis, individual tests are performed as well to test whether the difference between the distances traveled per drone by the RM and the DM are significant for each drone count. It is confirmed from the Kolmogorov-Smirnov test that the null hypothesis, which assumes that the distributions follow a normal distribution, can be rejected in all cases. Thus, none of the samples originate from a normal distribution. For this reason, the Mann-Whitney U test is used to test the null hypothesis: there is no difference between the distance flown distribution of the RM when compared to the DM. This test is repeated for each customer count, and therefore a Bonferroni correction has to be applied. Since there are 4 customer counts, 4 post-hoc tests are performed, and therefore the threshold significance level α should be corrected to $\alpha = \frac{0.05}{4} = 0.0125$. In every case, the p-value is found to be 0.00, resulting in the conclusion that the null hypothesis can be rejected in every of the four cases. There thus exists a significant difference between the distance that a drone travels with the DM when compared to the RM for every customer count.

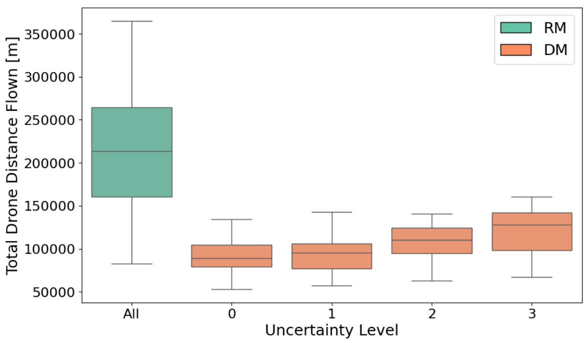
In addition to the distance flown per drone, the total drone distance for customer counts 10, 25, 50, and 100 are shown in Figure 6.1.



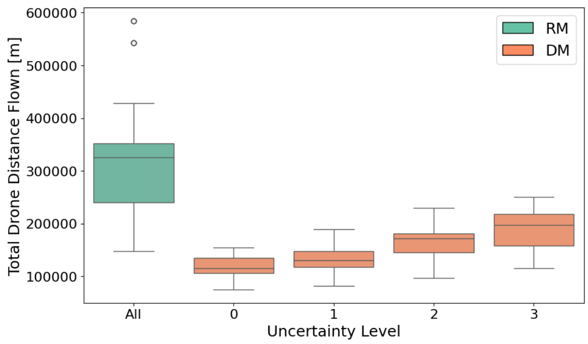
(a) Total Distance Flown for Customer Count 10



(b) Total Distance Flown for Customer Count 25



(c) Total Distance Flown for Customer Count 50



(d) Total Distance Flown for Customer Count 100

Figure 6.1: Drone Distances Flown per Uncertainty Level

References

[1] KPMG International. *The Truth About Online Consumers*. Tech. rep. KPMG International, Jan. 2017. URL: https://assets.kpmg.com/content/dam/kpmg/xx/pdf/2017/01/the-truth-about-online-consumers.pdf.

[2] Martin Barthel et al. "Winning Formulas for E-commerce Growth". In: *Boston Consulting Group* (Oct. 2023). Accessed: 2024-04-04. URL: https://www.bcg.com/publications/2023/winning-formulas-for-e-commerce-growth.

[3] Georgia Wells and Laura Stevens. *Amazon Conducts First Commercial Drone Delivery*. Accessed: 2024-04-04. 2016. URL: https://www.wsj.com/articles/amazon-conducts-first-commercial-drone-delivery-1481725956.

[4] David Pierce. "Delivery drones are coming: Jeff Bezos previews half-hour shipping". In: *The Verge* (2013). Accessed: 2024-04-04. URL: https://www.theverge.com/2013/12/1/5164340/delivery-drones-are-coming-jeff-bezos-previews-half-hour-shipping.

[5] Aishwarya Raghunatha et al. "Critical assessment of emissions, costs, and time for last-mile goods delivery by drones versus trucks". In: *Scientific Reports* 13.1 (July 21, 2023), p. 11814. DOI: 10.1038/s41598-023-38922-z. URL: https://doi.org/10.1038/s41598-023-38922-z.

[6] NBC News. *Google Wants to Deliver Your Stuff by Drone in 2017*. https://www.nbcnews.com/tech/tech-news/google-wants-deliver-your-stuff-drone-2017-n455861. Accessed: 2024-04-04. 2015.

[7] Walmart. *We're Bringing the Convenience of Drone Delivery to 4 Million U.S. Households in Partnership with DroneUp*. https://corporate.walmart.com/news/2022/05/24/were-bringing-the-convenience-of-drone-delivery-to-4-million-u-s-households-in-partnership-with-droneup. Accessed: 2024-04-04. May 2022.

[8] ABC News. *Amazon, UPS Drone Delivery Options*. https://abcnews.go.com/Technology/amazon-ups-drone-delivery-options/story?id=21086160. Accessed: 2024-04-04. Dec. 2013.

[9] DHL. *DHL Launches Its First Regular Fully Automated and Intelligent Urban Drone Delivery Service*. https://group.dhl.com/en/media-relations/press-releases/2019/dhl-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html. Accessed: 2024-04-04. 2019.

[10] The Verge. *Google's Alphabet Wing gains FAA approval for commercial drone deliveries*. https://www.theverge.com/2019/4/23/18512658/google-alphabet-wing-drone-delivery-service-faa-approval-commercial-deliveries. Accessed: 2024-04-04. Apr. 2019.

[11] CNBC. *Amazon Prime Air drone delivery fleet gets FAA approval*. https://www.cnbc.com/2020/08/31/amazon-prime-now-drone-delivery-fleet-gets-faa-approval.html. Accessed: 2024-04-04. Aug. 2020.

[12] Amazon UK. *Photos of Amazon's New Prime Air Drone*. https://www.aboutamazon.co.uk/news/operations/photos-of-amazons-new-prime-air-drone. Accessed: 2024-04-04. 2023.

[13] Niels Agatz et al. "Optimization Approaches for the Traveling Salesman Problem with Drone". In: *SSRN Electronic Journal* (Jan. 2015). DOI: 10.2139/ssrn.2639672.

[14] Chase C. Murray et al. "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery". In: *Transportation Research Part C: Emerging Technologies* 54 (2015), pp. 86–109. DOI: https://doi.org/10.1016/j.trc.2015.03.005. URL: https://www.sciencedirect.com/science/article/pii/S0968090X15000844.

[15] Karla Hoffman et al. "Traveling Salesman Problem". In: Jan. 2013, pp. 1573–1578. DOI: 10.1007/978-1-4419-1153-7_1068.

[16] Paolo Toth et al. *The vehicle routing problem*. SIAM, 2002.

[17] M.R. Garey et al. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Mathematical Sciences Series. Freeman, 1979. URL: https://books.google.nl/books?id=fjxGAQAAIAAJ.

[18] John C. Mankins. *Technology Readiness Levels*. Tech. rep. Accessed: 2024-04-04. NASA, 1995.

[19] Jacco M Hoekstra et al. "Bluesky ATC simulator project: an open data and open source approach". In: *Proceedings of the 7th international conference on research in air transportation*. Vol. 131. FAA/Eurocontrol Washington, DC, USA. 2016, p. 132.

[20] Ruowei Zhang et al. "A Review on the Truck and Drone Cooperative Delivery Problem". In: *Unmanned Systems* 12.4 (2024), pp. 1–25. DOI: 10.1142/S2301385024300014. eprint: https://doi.org/10.1142/S2301385024300014. URL: https://doi.org/10.1142/S2301385024300014.

[21] John Gunnar Carlsson et al. "Coordinated Logistics with a Truck and a Drone". In: *Management Science* 64.9 (2017), pp. 4052–4069.

[22] Paul Bouman et al. "Dynamic Programming Approaches for the Traveling Salesman Problem with Drone". In: *SSRN Electronic Journal* (Jan. 2017). DOI: 10.2139/ssrn.3035323.

[23] Xiaoshan Bai et al. "Efficient Package Delivery Task Assignment for Truck and High Capacity Drone". In: *IEEE Transactions on Intelligent Transportation Systems* PP (June 2023). DOI: 10.1109/TITS.2023.3287163.

[24] Sergio Ferrandez et al. "Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm". In: *Journal of Industrial Engineering and Management* 9 (Apr. 2016), p. 374. DOI: 10.3926/jiem.1929.

[25] Yong Sik Chang et al. "Optimal delivery routing with wider drone-delivery areas along a shorter truck-route". In: *Expert Systems with Applications* 104 (2018), pp. 307–317. DOI: https://doi.org/10.1016/j.eswa.2018.03.032. URL: https://www.sciencedirect.com/science/article/pii/S0957417418301775.

[26] Chase C. Murray et al. "The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones". In: *Transportation Research Part C: Emerging Technologies* 110 (2020), pp. 368–398. DOI: https://doi.org/10.1016/j.trc.2019.11.003. URL: https://www.sciencedirect.com/science/article/pii/S0968090X19302505.

[27] Zhihao Luo et al. "The Multi-visit Traveling Salesman Problem with Multi-Drones". In: *Transportation Research Part C: Emerging Technologies* 128 (2021), p. 103172. DOI: https://doi.org/10.1016/j.trc.2021.103172. URL: https://www.sciencedirect.com/science/article/pii/S0968090X2100190X.

[28] Phan Anh Tu et al. "Traveling Salesman Problem with Multiple Drones". In: *Proceedings of the 9th International Symposium on Information and Communication Technology*. SoICT '18. Danang City, Viet Nam: Association for Computing Machinery, 2018, pp. 46–53. DOI: 10.1145/3287921.3287932. URL: https://doi.org/10.1145/3287921.3287932.

[29] Ho Young Jeong et al. "Drone routing problem with truck: Optimization and quantitative analysis". In: *Expert Systems with Applications* 227 (2023), p. 120260. DOI: https://doi.org/10.1016/j.eswa.2023.120260. URL: https://www.sciencedirect.com/science/article/pii/S0957417423007625.

[30] Stefan Poikonen et al. "Multi-visit drone routing problem". In: *Computers & Operations Research* 113 (2020), p. 104802. DOI: https://doi.org/10.1016/j.cor.2019.104802. URL: https://www.sciencedirect.com/science/article/pii/S0305054819302448.

[31] Mohamed R. Salama et al. "Collaborative truck multi-drone routing and scheduling problem: Package delivery with flexible launch and recovery sites". In: *Transportation Research Part E: Logistics*

and Transportation Review 164 (2022), p. 102788. DOI: https://doi.org/10.1016/j.tre.2022.102788. URL: https://www.sciencedirect.com/science/article/pii/S1366554522001776.

[32] Bahare Mahmoudi et al. "Energy-constrained multi-visit TSP with multiple drones considering non-customer rendezvous locations". In: *Expert Systems with Applications* 210 (2022), p. 118479. DOI: https://doi.org/10.1016/j.eswa.2022.118479. URL: https://www.sciencedirect.com/science/article/pii/S0957417422015676.

[33] Jose Miguel Leon-Blanco et al. "A multi-agent approach to the truck multi-drone routing problem". In: *Expert Systems with Applications* 195 (2022), p. 116604. DOI: https://doi.org/10.1016/j.eswa.2022.116604. URL: https://www.sciencedirect.com/science/article/pii/S0957417422000975.

[34] Bao Tong et al. "Optimal Route Planning for Truck–Drone Delivery Using Variable Neighborhood Tabu Search Algorithm". In: *Applied Sciences* 12.1 (2022). DOI: 10.3390/app12010529. URL: https://www.mdpi.com/2076-3417/12/1/529.

[35] Dyutimoy Nirupam Das et al. "Synchronized Truck and Drone Routing in Package Delivery Logistics". In: *IEEE Transactions on Intelligent Transportation Systems* 22.9 (2021), pp. 5772–5782. DOI: 10.1109/TITS.2020.2992549.

[36] Patchara Kitjacharoenchai et al. "Multiple traveling salesman problem with drones: Mathematical model and heuristic approach". In: *Computers & Industrial Engineering* 129 (2019), pp. 14–30. DOI: https://doi.org/10.1016/j.cie.2019.01.020. URL: https://www.sciencedirect.com/science/article/pii/S0360835219300245.

[37] Felix Tamke et al. "A branch-and-cut algorithm for the vehicle routing problem with drones". In: *Transportation Research Part B: Methodological* 144 (2021), pp. 174–203. DOI: https://doi.org/10.1016/j.trb.2020.11.011. URL: https://www.sciencedirect.com/science/article/pii/S0191261520304410.

[38] Konstantin Kloster et al. "The multiple traveling salesman problem in presence of drone- and robot-supported packet stations". English. In: *European Journal of Operational Research* 305.2 (Mar. 2023). Funding Information: The authors wish to thank the anonymous reviewers for their valuable comments and recommendations on the original version of this study. Publisher Copyright: © 2022 Elsevier B.V., pp. 630–643. DOI: 10.1016/j.ejor.2022.06.004.

[39] Patchara Kitjacharoenchai et al. "Two echelon vehicle routing problem with drones in last mile delivery". In: *International Journal of Production Economics* 225 (2020), p. 107598. DOI: https://doi.org/10.1016/j.ijpe.2019.107598. URL: https://www.sciencedirect.com/science/article/pii/S0925527319304335.

[40] C.K.Y. Lin. "A vehicle routing problem with pickup and delivery time windows, and coordination of transportable resources". In: *Computers & Operations Research* 38.11 (2011), pp. 1596–1609. DOI: https://doi.org/10.1016/j.cor.2011.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0305054811000335.

[41] Quang Minh Ha et al. "On the min-cost Traveling Salesman Problem with Drone". In: *Transportation Research Part C: Emerging Technologies* 86 (2018), pp. 597–621. DOI: https://doi.org/10.1016/j.trc.2017.11.015. URL: https://www.sciencedirect.com/science/article/pii/S0968090X17303327.

[42] Ho Young Jeong et al. "Collaborative Hybrid Delivery System: Drone Routing Problem Assisted by Truck". In: *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems*. Springer International Publishing, 2021, pp. 33–42.

[43] Naifeng Wen et al. "Online UAV path planning in uncertain and hostile environments". In: *International Journal of Machine Learning and Cybernetics* 8.2 (2017), pp. 469–487. DOI: 10.1007/s13042-015-0339-4. URL: https://doi.org/10.1007/s13042-015-0339-4.

[44] Stefano Primatesta et al. "A Risk-Aware Path Planning Strategy for UAVs in Urban Environments". In: *Journal of Intelligent & Robotic Systems* 95.2 (2019), pp. 629–643. DOI: 10.1007/s10846-018-0924-3. URL: https://doi.org/10.1007/s10846-018-0924-3.

[45] Praveen Kumar Selvam et al. “Collision-free Path Planning for UAVs using Efficient Artificial Potential Field Algorithm”. In: *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)* (2021), pp. 1–5. URL: https://api.semanticscholar.org/CorpusID:235455310.

[46] Xiangpeng Wan et al. “Optimal Collision-Free Navigation for Multi-Rotor UAV Swarms in Urban Areas”. In: (2019), pp. 1–5. DOI: 10.1109/VTCSpring.2019.8746332.

[47] Ahmed Bahabry et al. “Low-Altitude Navigation for Multi-Rotor Drones in Urban Areas”. In: *IEEE Access* 7 (2019), pp. 87716–87731. DOI: 10.1109/ACCESS.2019.2925531.

[48] Marta J. Ribeiro et al. “Analysis of Conflict Resolution Methods for Manned and Unmanned Aviation Using Fast-Time Simulations”. In: *Proceedings of the 9th SESAR Innovation Days*. SESAR. Athens, Greece, Dec. 2–5, 2019. URL: http://resolver.tudelft.nl/uuid:73d8d050-9755-4dde-bf36-c6c0cc305f37.

[49] Jacco M Hoekstra et al. “Designing for safety: the ‘free flight’ air traffic management concept”. In: *Reliability Engineering & System Safety* 75.2 (2002), pp. 215–232. DOI: https://doi.org/10.1016/S0951-8320(01)00096-5. URL: https://www.sciencedirect.com/science/article/pii/S0951832001000965.

[50] Emmanuel Sunil et al. “Analysis of Airspace Structure and Capacity for Decentralized Separation Using Fast-Time Simulations”. In: *Journal of Guidance, Control, and Dynamics* 40.1 (2017), pp. 38–51. DOI: 10.2514/1.G000528. eprint: https://doi.org/10.2514/1.G000528. URL: https://doi.org/10.2514/1.G000528.

[51] E. Sunil et al. “Three-dimensional conflict count models for unstructured and layered airspace designs”. In: *Transportation Research Part C: Emerging Technologies* 95 (2018), pp. 295–319. DOI: 10.1016/j.trc.2018.05.031. URL: https://doi.org/10.1016/j.trc.2018.05.031.

[52] Zizhen Zhang et al. “Solving Dynamic Traveling Salesman Problems With Deep Reinforcement Learning”. In: *IEEE Transactions on Neural Networks and Learning Systems* 34.4 (2023), pp. 2119–2132. DOI: 10.1109/TNNLS.2021.3105905.

[53] Michalis Mavrovouniotis et al. “Ant Colony Optimization With Local Search for Dynamic Traveling Salesman Problems”. In: *IEEE Transactions on Cybernetics* 47.7 (2017), pp. 1743–1756. DOI: 10.1109/TCYB.2016.2556742.

[54] C. Archetti et al. “Dynamic traveling salesman problem with stochastic release dates”. In: *European Journal of Operational Research* 280.3 (2020), pp. 832–844. DOI: https://doi.org/10.1016/j.ejor.2019.07.062. URL: https://www.sciencedirect.com/science/article/pii/S0377221719306411.

[55] OpenStreetMap contributors. *Planet dump retrieved from https://planet.osm.org*. https://www.openstreetmap.org. 2017.

[56] Geoff Boeing. *Modeling and Analyzing Urban Networks and Amenities with OSMnx*. 2024.