

**TECHNISCHE UNIVERSITÄT DORTMUND**

**Fakultät Maschinenbau**

**IT in der Produktion und Logistik (ITPL)**



**Bachelorarbeit**

**Travelling-Salesman-Untersuchung beim Verbund von Drohnen und**

**Lastkraftwagen in Dortmund**

Betreuer: Dr.-Ing. Dipl.-Inform. Anne Antonia Scheidler  
M.Sc. Florian Hochkamp

Vorgelegt von: Till Feldkamp

Matrikelnummer: 197399

Studiengang: Bachelor of Science Wirtschaftsingenieurwesen

Ausgegeben am: 06.04.2021

Eingereicht am: 29.06.2021

Dortmund, im Juni 2021

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis .....</b>	<b>I</b>
<b>Tabellenverzeichnis .....</b>	<b>II</b>
<b>Abkürzungsverzeichnis .....</b>	<b>III</b>
<b>1. Einleitung .....</b>	<b>1</b>
<b>2. Logistische Grundlagen für die Paketauslieferung mittels Unmanned-Aerial-Vehicles.....</b>	<b>3</b>
2.1 Letzte Meile Konzepte .....	3
2.2 Aufbau, Arten und Einsatz von Unmanned-Aerial-Vehicles .....	5
2.3 Technische Limitierungen und externe Restriktionen .....	6
<b>3. Theoretische Grundlagen der Simulation und des Travelling-Salesman-Problems .....</b>	<b>9</b>
3.1 Grundlagen der Simulation und des Modells .....	9
3.2 Grundlagen eines Travelling-Salesman-Problems.....	14
3.3 Travelling-Salesman-Probleme für Unmanned-Aerial-Vehicles .....	16
<b>4. Theoretische Entwicklung der Paketauslieferung mittels Unmanned-Aerial-Vehicles und Lastkraftwagen in Dortmund.....</b>	<b>22</b>
4.1 Aufbau des Simulationsmodells.....	22
4.2 Spezialisierung des Modells auf Dortmund.....	24
<b>5. Simulation der Paketauslieferung mittels Unmanned-Aerial-Vehicles und Lastkraftwagen in Dortmund.....</b>	<b>25</b>
5.1 Exemplarische Implementierung des Modells.....	25
5.2 Simulationsdurchführung und Erhalt der Ergebnisse .....	29
5.3 Auswertung der Ergebnisse.....	32
5.4 Fazit .....	34
<b>6. Zusammenfassung und Ausblick .....</b>	<b>35</b>
<b>Literaturverzeichnis .....</b>	<b>36</b>
<b>Anhang.....</b>	<b>39</b>

# Abbildungsverzeichnis

Abbildung 1: Wege um ein System zu Studieren.....	10
Abbildung 2: Schematischer Verlauf der Kosten von der Planung bis zur Inbetriebnahme.....	11
Abbildung 3: Vorgehensmodell bei der Simulation mit V&V.....	12
Abbildung 4: Beispiel eines FSTSP und PDSTSP.....	17
Abbildung 5: Beispiel des mFSTSP mit einem oder drei UAVs im Vergleich zur herkömmlichen Zustellung mittels Zustellfahrzeug.....	18
Abbildung 6: Schematischer Verlauf der Datenerhebung.....	26
Abbildung 7: Beispiel einer Visualisierung in QGIS von 20 Kunden und der Routen.....	27
Abbildung 8: Exemplarischer Programmablaufplan von gen_csv_files.py.....	28
Abbildung 9: Gegenüberstellung der Auslieferungsvarianten im innerstädtischen Bezirk.....	30
Abbildung 10: Gegenüberstellung der Auslieferungsvarianten im Vorort.....	30

# Tabellenverzeichnis

Tabelle 1: Berechnete Zeit-Ersparnisse bei 8 Kunden .....	31
Tabelle 2: Berechnete Zeit-Ersparnisse bei 20 Kunden .....	31
Tabelle 3: Berechnete Zeit-Ersparnisse bei 50 Kunden .....	31
Tabelle 4: Berechnete Einsparung der Kilometer für 8 Kunden .....	32
Tabelle 5: Beispiel der Input-Datei des gen_csv_files.py Programms.....	39
Tabelle 6: Beispiel der ersten Output-Datei des gen_csv_files.py Programms .....	39
Tabelle 7: Beispiel der zweiten Output-Datei des gen_csv_files.py Programms.....	39
Tabelle 8: Simulationsläufe und Argumente.....	40

# Abkürzungsverzeichnis

BMVI	Bundesministerium für Verkehr und digitale Infrastruktur
EASA	European-Union-Aviation-Safety-Agency
FSTSP	Flying-Sidekick-TSP
GPS	Global-Positioning-System
Lidar	Light-Detection-And-Ranging
LP	Lineare Programmierung
mFSTSP	multiple-Flying-Sidekick-TSP
MILP	Gemischt-ganzzahlige Programmierung
NP-vollständig	nichtdeterministisch in Polynomialzeit - vollständig
ORDBMS	objektrelationales Datenbankmanagementsystem
OSM	Open-Street-Map
PDSTSP	Parallel-Drone-Scheduling-TSP
QP	Quadratische Programmierung
TSP	Travelling-Salesman-Problem
TSP-D	TSP-With-Drone
UAV	Unmanned-Aerial-Vehicles
VDI	Verein Deutscher Ingenieure

# 1. Einleitung

Drohnen gewinnen in der heutigen Gesellschaft durch ihren Nutzen und ihre Einsatzmöglichkeiten zunehmend an Bedeutung (Dorling et al., 2017). Dadurch, dass Drohnen den Luftweg nutzen, um zu ihrem Ziel zu kommen, bringen sie der Logistikbranche ganz neue Möglichkeiten, Waren zu transportieren. Im Sinne der Industrie 4.0 und der damit verbundenen Digitalisierung werden Drohnen, oder auch Unmanned-Aerial-Vehicle (UAV) genannt, unter anderem für die Inventur in Hochregallagern eingesetzt (ten Hompel et al., 2020). Darüber hinaus kommen UAVs in Bereichen wie der Energie-, Land- und Forstwirtschaft, im Bauwesen und in der Filmindustrie zum Einsatz (Rojas Vilorio et al., 2021). Der Einsatz von UAVs zur Paketauslieferung steht noch bevor und wird zurzeit durch wissenschaftliche Arbeiten erforscht (Otto et al., 2018).

Die Anwendung von UAVs im öffentlichen Sektor unterliegt technischen Limitierungen und verschiedenen externen Restriktionen, wie beispielsweise Gesetze und Wetterbedingungen (Hänsenberger, 2018). Unternehmen der Paketzustellungsbranche forschen dennoch daran, wie die Zustellung von Paketen mittels UAVs möglich gemacht werden kann (Dorling et al., 2017). Ein Ziel von Unternehmen in der Logistik ist es den letzten Schritt der Zustellung zum Kunden, auch letzte Meile genannt, möglichst effizient mittels UAVs zu gestalten (Boysen et al., 2021). Eine Optimierungsmöglichkeit, um dieses Problem effizienter anzugehen, ist das Travelling-Salesman-Problem (TSP). Klassischerweise bezieht sich ein TSP nur auf die Streckenoptimierung einer Handelsreise oder einer Lieferung (Dantzig et al., 1954). In dem erweiterten TSP, welches in dieser Arbeit als Grundlage für die Simulation dient, werden jedoch ein Lastkraftwagen und UAVs zusammen betrachtet, sodass es möglich ist, Pakete parallel oder synchron ausliefern zu können. Daher muss das klassische TSP dementsprechend neugestaltet werden. Dies wurde 2015 von Murray und Chu bereits thematisiert. Sie entwickelten ein TSP für die synchrone Auslieferung von Paketen durch ein UAV und einem Lastkraftwagen und eines für die parallele Auslieferung (Murray & Chu, 2015). Dieses TSP wurde im Jahr 2020 erweitert, indem mehrere UAVs von einem LKW aus starten können (Murray & Raj, 2020). Da es zu diesem Stand der Entwicklung von UAVs noch nicht möglich ist, dieses Verfahren experimentell zu erproben und sie technischen Limitierungen unterliegen, wird in dieser Arbeit eine Simulation des Verfahrens betrachtet. Simulationen werden in der Logistik eingesetzt, um Erkenntnisse zu erlangen, die auf das reale System anzuwenden sind (Gutenschwager et al., 2017). Dadurch können im Vorfeld Probleme und Verbesserungen in der Zustellung von Paketen mittels UAVs erkannt und behoben werden, welche in der Schlussfolgerung bewertet werden und es zu veränderten Zielen und Hypothesen kommen kann (Rabe et al., 2008).

Das Ziel dieser Arbeit ist die Modellimplementierung der Simulation eines TSPs der Paketzustellung im Verbund von UAVs und Lastkraftwagen in Dortmund. Dabei wird geprüft, ob der Einsatz von mehreren UAVs mit einem Lastkraftwagen, im Sinne des multiple-Flying-Sidekick-TSP (mFSTSP), möglich ist. Des Weiteren soll die Dauer der Zustellung minimiert werden. Die Stadt Dortmund wird für die Simulation exemplarisch gewählt, da sie eine höhere Bevölkerungsdichte im innerstädtischen Bezirk als in den umliegenden Vororten aufweist. Somit kann festgestellt werden welche Auswirkung die Bevölkerungsdichte auf das aufgestellte Modell hat. Für die Simulation, werden vorab alle nötigen Grundlagen zusammengetragen, die eine Paketzustellung in Dortmund mittels UAVs möglich machen. Innerhalb der Simulation müssen jedoch die Annahmen getroffen werden, dass es den UAVs bereits technisch möglich ist ein Paket zum Empfänger zuzustellen und die Limitierungen sowie externen Restriktionen

nicht mehr vorliegen. Es wird sich auf technische Werte existierender UAVs in diesen Annahmen bezogen. Eine detailliertere Aufstellung der Annahmen erfolgt in Kapitel 4. Zur exemplarischen Implementierung des Modells wird die Programmiersprache Python verwendet. Durch die Ergebnisse der Simulation soll beurteilt werden, ob die Zustellung durch UAVs in Kombination mit Lastkraftwagen eine Alternative zur herkömmlichen Auslieferung, mittels Paketboten und Zustellfahrzeug, darstellt.

Es werden in Kapitel 2 und 3 die logistischen und theoretischen Grundlagen dargelegt. In diesen Kapiteln werden relevante Theorien für die spätere Konzeptentwicklung und Simulation zusammengetragen. Die logistischen Grundlagen lassen sich in Letzte-Meile-Konzepte, Aufbau und Einsatz von UAVs und in technische Limitierungen unterteilen. Die Vorstellung der Letzte-Meile-Konzepte kann einen Überblick und eine Einordnung verschaffen, inwiefern die Auslieferung mittels UAV und Lastkraftwagen im Vergleich zu den anderen Konzepten einzuordnen ist. Des Weiteren wird der Aufbau und die verschiedenen Einsatzmöglichkeiten aufgezeigt, um das Potenzial von UAVs zu veranschaulichen. Jedoch auch um Einschränkungen offen zu legen, die die Modellbildung beeinflussen. Aufgrund dessen wird sich mit den technischen Limitierungen und externen Restriktionen beschäftigt, damit passende Annahmen für das Modell getroffen werden können. Die theoretischen Grundlagen werden benötigt, um den Ablauf und die Funktion einer Simulation darzustellen. Nachfolgend wird eine Simulation für die Paketauslieferung Mittels UAV und Lastkraftwagen angewendet. Demnach wird auch die Theorie eines TSPs und speziell die des erweiterten TSP für UAVs, da dieses Bestandteil der Simulation sein wird, benötigt.

In Kapitel 4 wird schließlich das Konzept des Modells aufgestellt und die zuvor genannten Grundlagen werden genutzt, um das Simulationsmodell auf die betrachtete Zielsetzung anzupassen. Eine Spezialisierung des Modells auf Dortmund ist notwendig, da das angenommene Modell ohne geographische Daten und Annahmen von Dortmund, nicht funktioniert. Nach der Konzeptentwicklung wird in Kapitel 5 der Ablauf der Datenerhebung und das Generieren von zwei benötigten CSV-Dateien deutlich gemacht. Diese sind Voraussetzung für das Verwenden des angenommenen Modells. Nach Beschreibung des Ablaufs der Datenerhebung kann die Simulation durchgeführt und Ergebnisse der Simulation ausgewertet werden. Abschließend werden die erlangten Ergebnisse in einem Fazit zusammengetragen und ein Ausblick in Bezug auf die Auslieferung von Paketen mittels UAVs gegeben. Zusätzlich wird beschrieben, wie das für Dortmund aufgestellte Modell mit wenigen Schritten verändert werden muss, dass es auf andere Städte anzuwenden ist.

## 2. Logistische Grundlagen für die Paketauslieferung mittels Unmanned-Aerial-Vehicles

In diesem Abschnitt werden die logistischen Grundlagen für die Paketauslieferung mittels UAVs in Dortmund zusammengefasst. Zunächst werden verschiedene Letzte-Meile-Konzepte vorgestellt, um Alternativen zur Auslieferung mittels UAVs dazulegen. Daraufhin werden wesentliche Eigenschaften von UAVs aufgezeigt, die später in der Simulation Anwendung finden, wie zum Beispiel der qualitative Aufbau von UAVs und die verschiedenen Einsatzmöglichkeiten. Zudem wird eine Übersicht der technischen Limitierungen und externen Restriktionen erstellt.

### 2.1 Letzte Meile Konzepte

Die letzte Meile ist ein Begriff in der Logistik, die den finalen Zustellprozess zum privaten Kundenhaushalt in städtischen und ländlichen Gebieten beschreibt (Aurambout et al., 2019; Boysen et al., 2021; Rojas Vilorio et al., 2021). Die Zustellung innerhalb der letzten Meile beginnt erst dann, wenn das Paket meist nach einem Langstreckentransport in einem örtlichen zentralen Depot angekommen ist (Boysen et al., 2021). An diesem Depot werden alle Pakete nach der Zustelladresse Bezirken zugeteilt. Ist die Aufteilung abgeschlossen, werden die Pakete auf der letzten Meile zum Kunden zugestellt. Je weniger Pakete in einem Lastkraftwagen gleichzeitig transportiert werden, desto mehr Kosten entstehen für das Unternehmen. Daher ist der letzte Schritt der Zustellung, in dem nur noch kleinere Zustellfahrzeuge benutzt werden, mit höheren Kosten verbunden. Dazu kommt die Häufigkeit von Fehlzustellungen, die meist einen Umweg oder eine weitere Zustellung zur Folge hat (Aurambout et al., 2019).

Die letzte Meile wird von Boysen et al. (2021) in verschiedene Herausforderungen aufgeteilt. Laut Boysen müssen sich Unternehmen, welche die Effizienz innerhalb der letzten Meile steigern wollen, mit der *Nachhaltigkeit* beschäftigen. Dies kann zum Beispiel der Umstieg auf elektrische Fahrzeuge sein. Zudem sollten die entstehenden *Kosten* für das Unternehmen verringert werden. Laut einer Studie belaufen sich die Kosten in einem Unternehmen, für die Zustellung mittels Zustellfahrzeug und Paketboten, je nach Kundendichte zwischen zwei und sechs Euro (Boysen et al., 2021). Die weltweit *steigende Anzahl* der Onlinebestellungen, die laut Boysen et al. (2021) einen Zuwachs von 23,3 Prozent erfährt, stellt eine Herausforderung für die Logistikunternehmen da. Dazu kommt die Häufigkeit der Fehlzustellung, aufgrund von nicht anzutreffende Kunden, von zwölf bis 60 Prozent (Boysen et al., 2021). Eine weitere Herausforderung für Unternehmen ist der *Zeitdruck*. Immer mehr Online-Händler werben mit Sameday- oder Nextday-Paketzustellung (Zustellung am selben oder am Folgetag), wodurch die Logistikunternehmen unter Zeitdruck geraten. Dieser Zeitdruck nimmt bei Saisonschlussverkäufen weiter zu. Die letzte Herausforderung, die Boysen nennt, ist die der *älter werdenden Arbeitnehmer* in dieser Branche. Es werden im Verhältnis zu wenig junge Arbeitnehmer eingestellt, was auf lange Sicht zu einer älter werdenden Belegschaft führt. Die schneller in Rente gehen und irgendwann Mangel an Arbeitnehmern herrschen kann. Daher ist für Unternehmen eine automatische Zustellung zukunftsversprechend (Boysen et al., 2021).



Die Letzte-Meile-Konzepte werden nach Boysen et al. (2021) in bereits angewandte Konzepte und in nicht angewendete Konzepte gegliedert. Dabei werden die genannten Herausforderungen geprüft, ob sie Zutreffen für das jeweilige Letzte-Meile-Konzept oder nicht. Zunächst werden die angewandten Konzepte erläutert:

- *Zustellfahrzeug*: Bei diesem Konzept werden die Zustellfahrzeuge an einem Depot mit Paketen beladen, welche die gleiche Zustellregion aufweisen. Im Hinblick auf die genannten Herausforderungen kann dieses Konzept durch die Nutzung von elektrischen Fahrzeugen nachhaltiger umgesetzt werden. Jedoch sind die anderen Herausforderungen zur Effizienzsteigerung bei diesem Konzept schwer umzusetzen. Der Paketbote muss jedes einzelne Paket selbst zustellen und ist daher für eine alternde Belegschaft nicht nachhaltig. Zudem steigen ungeplante Kosten dieses Konzepts aufgrund von Fehlzustellungen und der damit verbundenen zweiten Zustellung schnell an. Ein weiterer Nachteil von Zustellfahrzeugen ist es, dass das Verkehrsaufkommen in den Innenstädten ansteigt.
- *Cargo-Bikes*: Ein anderes Konzept ist die Zustellung mit elektrischen Cargo-Bikes. Dieses Konzept wird verstärkt in Stadtzentren benutzt (Gruber et al., 2014). Der Zustellprozess von Cargo-Bikes kann nach Boysen et al. (2021) folgendermaßen beschrieben werden, zunächst werden Micro-Depots durch Lastkraftwagen des normalen Depots beliefert. Von diesen Micro-Depots aus können die Cargo-Bikes beladen und die Pakete zugestellt werden. Ein Vorteil gegenüber normalen Zustellfahrzeugen ist, dass die Cargo-Bikes zugangsbeschränkte Bereiche und auch gegen die eigentliche Fahrtrichtung von Einbahnstraßen fahren dürfen und eine bessere Parksituation bieten. Ein Nachteil der Cargo-Bikes ist hingegen, dass diese aufgrund der geringeren Ladefläche mehrmals beladen werden und zum Micro-Depot zurückkehren müssen. Nach Boysen werden Kosten, Emissionen, Lärm und Staus um 40 Prozent reduziert durch die Nutzung von Cargo-Bikes (Boysen et al., 2021). Die Herausforderungen der letzten Meile werden hinsichtlich der Nachhaltigkeit aufgrund der elektrisch betriebenen Cargo-Bikes gedeckt.
- *Selbstbedienung*: Das letzte Konzept, das heutzutage bereits genutzt wird, ist die Selbstbedienung. Bei dieser Art und Weise der Zustellung akzeptieren die Kunden, dass sie ihr Paket in einem Schließfach oder einem Paketshop nach Zustellung selbst abholen. Der Zustellprozess ist demnach vom Depot aus nur noch zu den jeweiligen Schließfächern oder Paketshops notwendig, bei denen eine Zustellung aufgrund der Öffnungszeiten des Paketshops oder dem Zugang zu den Schließfächern gewährleistet ist. So entstehen keine Kosten aufgrund nicht zugestellter Pakete. Je mehr Kunden diesen Service in Anspruch nehmen, desto weniger Kosten entstehen für das liefernde Unternehmen. Demnach werden durch diese Zustelloption die Herausforderung von Kosten, Zeitdruck, sowie die alternde Belegschaft berücksichtigt (Boysen et al., 2021).

Im Folgenden werden die noch nicht angewendeten Letzte-Meile-Konzepte genannt:

- *Zustellungsroboter*: Eins der zwei noch nicht angewendeten Konzepte sind autonome Zustellungsroboter. Solche Roboter können Pakete mit einem Gesamtgewicht von bis zu zehn Kilogramm transportieren und benutzen zur Zustellung den Bürgersteig, auf dem sie eine Geschwindigkeit von bis zu sechs Kilometer pro Stunde erreichen. Sie könnten entweder gleichzeitig mit einem Zustellfahrzeug Pakete zustellen oder von einem Depot aus, ihre Routen starten. Die Herausforderungen hinsichtlich Nachhaltigkeit und der alternder Belegschaft werden somit berücksichtigt (Boysen et al., 2021).

- *UAVs*: Das Konzept zur Zustellung mittels UAVs ist unter den noch nicht angewendeten Konzepten einzuordnen. UAVs sind Unbemannte Luftfahrzeuge, die im privaten Raum „Drohne“ genannt werden. Weitere Eigenschaften und den Aufbau von UAVs werden in Kapitel 2.2 näher erläutert. Die Zustellung läuft dabei autonom ab und wird entweder in Zusammenarbeit mit einem Zustellfahrzeug durchgeführt, auf dem das UAV starten und landen kann, oder vom Depot aus gestartet (Murray & Chu, 2015). Detailliertere Angaben zu den möglichen Zustellprozessen mittels UAV werden in Kapitel 3.3 erläutert. Das UAV kann nur ein Paket mit begrenztem Gewicht pro Zustellung mit sich führen und muss nach jeder Zustellung zum Zustellfahrzeug oder Depot zurückkehren. Das Maximalgewicht der Pakete variiert je nach Tragelast des UAV. Laut Vincent und Gartenberg (2019), kann das von Amazon eingesetzte UAV bis zu 2,5 Kilogramm schwere Pakete zustellen, was auf 75 bis 90 Prozent der Pakete von Amazon zutreffen soll. Die Zustellung durch das UAV über den Luftraum erfolgt ohne Hindernisse durch das örtliche Straßennetz, ist unabhängig vom Straßenverkehr und kann dadurch schnell durchgeführt werden. Wie sich der Luftraum durch mehrere UAVs verhält und somit sich auf die Zustellung auswirkt ist nicht bekannt. Das UAV verbessert den städtischen Straßenverkehr dadurch, dass weniger Verkehrsbelastung durch Zustellfahrzeuge in den Innenstädten herrscht. Die Herausforderungen der Nachhaltigkeit, Zeitdruck und der alternden Belegschaft werden durch die Zustellung mittels UAVs gedeckt (Boysen et al., 2021).

Durch die autonome Zustellung durch UAVs ist dieses Letzte-Meile-Konzept für Logistikunternehmen vielversprechend, da laut Boysen et al. (2021), die Unternehmen eine automatische Zustellung für die Zukunft bevorzugen werden. Der Nachteil, dass nur ein Paket mit limitiertem Gewicht pro Zustellung zugestellt werden kann, wird durch die Nutzung von mehreren UAVs, minimiert (Murray & Raj, 2020). Im folgenden Kapitel werden die Grundlagen zu UAVs wiedergegeben, bei denen es sich unter anderem um den qualitativen Aufbau sowie die verschiedenen Einsatzmöglichkeiten handelt.

## 2.2 Aufbau, Arten und Einsatz von Unmanned-Aerial-Vehicles

Unmanned-Aerial-Vehicle heißt übersetzt Unbemanntes Luftfahrzeug. Unbemannt heißt in diesem Fall, dass sich kein Pilot in dem Luftfahrzeug befindet. Entweder kann es durch einen Piloten am Boden per Fernsteuerung gesteuert werden, oder vollkommen autonom ihre Routen abfliegen (Rojas Vilorio et al., 2021). Die UAVs haben unterschiedliche Designs und Bauformen, können entweder feste Flügel haben, wie bei einem Flugzeug, oder sie werden als ein Drehflügler gebaut. Das UAV als Drehflügler besitzt meist vier oder mehr Rotoren, was eine bessere Stabilität während des Flugs in der Luft gewährleistet. Aufgrund der Anzahl der Rotoren werden diese Drehflügler auch Multicopter genannt. Sie generieren Auftrieb durch eine dynamische Verdrängung von Luft, die durch angeströmte profilierte Flächen, wie zum Beispiel Tragflächen oder Rotoren, entsteht (Mensen, 2013). Multicopter haben die positive Eigenschaft, dass sie vertikal Starten und Landen sowie in der Luft schweben können. Zusätzlich ist es den Multicoptern möglich scharfe Kurven zu fliegen, was UAVs mit festen Flügeln nicht können. Dies ist besonders gut geeignet in städtischen Gebieten, in denen wenig Platz und viel Verkehr vorherrscht (Otto et al., 2018). Außerdem sind sie im Vergleich zu den UAVs mit festen Flügeln einfacher und kostengünstiger herzustellen (Heutger & Kückelhaus, 2014). UAVs mit festen Flügeln brauchen zum Starten

und Landen eine Start- und Landebahn (Heutger & Kückelhaus, 2014). Jedoch ist die Ausdauer und Flugdistanz von UAVs mit festen Flügeln höher als die von Multicoptern, so Heutger und Kückelhaus (2014).

Die Technologie des UAVs entwickelt sich rasant und wird die Industrie in den nächsten Jahren revolutionieren (Otto et al., 2018). Nach aktuellen Prognosen soll der kommerzielle Markt für UAVs in Deutschland in den Jahren von 2020 bis 2025 um mehr als das doppelte ansteigen (Statista 2021a). Der Andrang an E-Commerce in Deutschland stieg in den letzten 10 Jahren um 279 Prozent an (Statista 2021b). Dies bleibt bei den führenden Logistikunternehmen nicht unbemerkt. Die Unternehmen haben größtenteils das Ziel die Letzte-Meile effizienter zu gestalten. Das Ziel und die damit verbundenen Herausforderungen lassen sich in Nachhaltigkeit, Kosten, Zeitdruck sowie der Belastung der alternden Arbeitnehmer zusammenfassen (Boysen et al., 2021). Daher ist die Paketzustellung mittels UAV laut Boysen et al. (2021) für diese Unternehmen aussichtsreich, da mit diesen UAVs der Großteil dieser genannten Faktoren verbessert werden können. Daran ist zu erkennen, dass der Nutzen von UAVs und die Einsatzfähigkeiten steigen. UAVs werden in unterschiedlichen Bereichen eingesetzt. In der Agrarwirtschaft werden sie benutzt, um 3D-Ansichten von Feldern anzulegen und um Felder mit Pestiziden zu besprühen. Sie können als mobile schwebende Wi-Fi Router in Katastrophengebieten genutzt werden, damit für die Rettungskräfte notwendige Systeme funktionieren. Ein Beispiel hierfür ist ein UAV, welches in einem Katastrophengebiet nach Überlebenden sucht und dafür zusätzlich eine Wi-Fi UAV benötigt (Rojas Vilorio et al., 2021). Sowie als UAV zur Zustellung von Paketen eingesetzt. Um den Nutzen und die Anwendungsmöglichkeiten weiter zu erhöhen muss sich mit den technischen Limitierungen und externen Restriktionen von UAVs auseinandergesetzt werden.

## 2.3 Technische Limitierungen und externe Restriktionen

Dem UAV liegen, bis zur autonomen Zustellung von Paketen, noch vermehrt technische Limitierungen sowie externe Restriktionen im Weg (de-Miguel-Molina & Santamarina-Campos, 2018; Hänsenberger, 2018; Otto et al., 2018). Die technischen Limitierungen lassen sich in Sicherheitstechnik, Batterieleistung und Flug-Limitierungen aufteilen. Die externen Restriktionen können in gesetzliche und ethische Restriktionen gegliedert werden. Durch ethische Restriktionen können rechtliche Restriktionen entstehen (de-Miguel-Molina & Santamarina-Campos, 2018).

Die UAVs müssen für die Paketauslieferung teils weite Strecken zurücklegen. Dadurch, dass UAVs Batterien als Energiequelle benutzen, ist die Batterieleistung ein wichtiger Faktor für UAVs. Durch die Entwicklung von Lithium Polymer Batterien, konnte im Vergleich zu alternativen Batterien, die Flugzeit von UAVs erhöht werden (Dorling et al., 2017). Der Energieverbrauch der UAVs hängt unter anderem von dem Gewicht des Pakets, des UAV selbst und der Batterie zusammen (Otto et al., 2018). Je schwerer das UAV wird, desto mehr wirkt sich das Gewicht auf die Flugzeit aus. Otto et al. (2018) zählt noch weitere Faktoren die sich auf den Energieverbrauch auswirken, auf. Nicht nur das Gewicht ist entscheidend für die Flugdauer, sondern auch der Typ des UAV und die Flughöhe. Hat das UAV feste Flügel wie ein Flugzeug, kann es besser im Geradeausflug gleiten und verbraucht somit weniger Energie. Hat sie jedoch Drehflügel bleibt der Vorteil des Gleitens aus. Je höher das UAV fliegt, desto dünner wird die Luft und die Propeller müssen sich schneller drehen, damit es genug Auftrieb generiert und nicht an

Höhe verliert. Auch die Wetterbedingungen können sich auf die Flugdauer auswirken. Bei starkem Gegenwind muss das UAV gegensteuern und verbraucht damit mehr Energie als bei windstillem Wetter (Otto et al., 2018). Ein Nachteil von Batterien im Gegensatz zu Verbrennungsmotoren ist die limitierte Anzahl an Ladezyklen. Batterien verlieren mit der Zeit an Leistung und müssen ausgetauscht werden (Dorling et al., 2017). Dadurch, dass die Flugdauer von diesen vielen Faktoren abhängt und variiert, kann die Flugreichweite ebenso variieren. In der Literatur finden sich verschiedene Angaben zu der maximalen Flugreichweite von UAVs. In Vincent und Gartenberg (2019) steht, dass Amazon eine Reichweite für ihr UAV von 15 Meilen bei einem Paketgewicht von fünf Pfund angibt. Das entspricht ungefähr 24,1 Kilometer bei einem Paketgewicht von 2,5 Kilogramm. Eine andere Quelle besagt, dass es UAVs möglich ist 15 Kilometer weit mit einem Paketgewicht von drei Kilogramm fliegen zu können (Otto et al., 2018).

Wie bereits erwähnt, treten ethische und rechtliche Probleme bei der Nutzung von UAVs zur Paketauslieferung auf. Den Bürgern kommen ethische Zweifel, wenn berücksichtigt wird, dass sich UAVs mit Paketgewichten von bis zu drei Kilogramm über die Köpfe von Bürgern hinwegbewegen (de-Miguel-Molina & Santamarina-Campos, 2018). Es treten Fragen auf wer die Verantwortung bei einem Unfall trägt und wer die Sicherheit eines autonomen UAV garantieren kann. Solche Fragen können nur durch eine rechtliche Grundlage für UAVs geklärt werden. Diese Gesetzesgrundlage wird durch die European Union Aviation Safety Agency (EASA) beschlossen. Um die Gesetze und die Folgen dieser, für diese Arbeit aufgreifen zu können, wird sich auf die von der EASA erstellten „Easy-Access-Rules“ und dem Artikel des Bundesministeriums für Verkehr und digitale Infrastruktur (BMVI) bezogen. Erst am 31.12.2020 trat ein neuer Beschluss der EASA in Kraft. In diesem Beschluss werden UAVs in drei Betriebskategorien eingeteilt: die offene, die spezielle und die zulassungspflichtige Kategorie (BMVI, 2021). Die offene Kategorie betrifft UAVs mit einem Startgewicht von 25 Kilogramm. Diese UAVs der offenen Kategorie dürfen nur innerhalb der Sichtweite der steuernden Person fliegen und eine maximale Höhe von 120 Metern erreichen. Der Transport von gefährlichen Gütern und der Abwurf dieser ist untersagt (EASA, 2019). Die spezielle Kategorie greift dann, wenn das UAV dem Einsatzspektrum der offenen Kategorie übersteigt. Dies wäre bei einem Flug außerhalb der Sichtweite oder bei einem Startgewicht von mehr als 25 Kilogramm der Fall (EASA, 2019). Die zulassungspflichtige Kategorie umfasst den Betrieb von großen und schweren UAVs, die zum Beispiel Menschen, gefährliche Güter oder Pakete für die Zustellung zum Endkunden transportieren. Diese Kategorie ist mit dem höchsten Risiko verbunden, daher wird der Sicherheitsaspekt wie bei der bemannten Luftfahrt behandelt. Alle UAVs dieser Kategorie müssen zertifiziert sein und der Betreiber solcher UAVs braucht eine von einer amtlichen Behörde ausgestellten Luftverkehrsbetreibergenehmigung. Zusätzlich muss der Fernpilot dieser UAVs eine Pilotenlizenz besitzen. In naher Zukunft wird jedoch auf autonome UAVs umgestellt und ein Fernpilot ist nicht mehr notwendig. Die EASA muss für die Umsetzung dieser zulassungspflichtigen Kategorie und den damit verbundenen Risiken fast alle geltenden Luftfahrtvorschriften neu aufstellen (EASA, 2019).

Um den hohen Sicherheitsaspekt zu gewährleisten versuchen die Hersteller die Sicherheitstechnik ihrer UAVs zu verbessern. Die geringe Flughöhe und das Auftreten von möglichen Hindernissen im Luftraum sowie im Tiefflug oder bei der Landung erfordern eine Hinderniserkennung zur Unfallvermeidung. Dafür braucht das UAV eine Echtzeit-Positionserfassung sowie eine Technik die Hindernisse frühzeitig erkennen kann und die UAVs an diesem Hindernis vorbei navigiert. Um den genauen Standort von UAVs zu bestimmen, werden das Global-Positioning-System (GPS) und weitere

Lokalisierungstechniken genutzt (Dorling et al., 2017). Für die Hinderniserkennung gibt es verschiedene Techniken. Zum Beispiel wird das Light-Detection-And-Ranging-System (Lidar) genutzt. Das Lidar-System wirft einen Laserstrahl in die Umgebung und kann durch Messung der Lichtlaufzeit die genaue Position von Hindernissen bestimmen. Das Lidar-System wird auch für das Aufzeichnen von Landschaften und das Suchen von Menschen, Tieren und anderen Lebewesen oder Gegenstände in Katastrophengebieten genutzt (Merz & Kendoul, 2013). Es werden auch bildverarbeitungs-basierte Techniken zur Hinderniserkennung eingesetzt. Bei dieser Technik wird ein teleoperiertes Navigationssystem für das UAV eingesetzt. Dieses basiert auf einer 3D-Karte der Umgebung, die mit Bildern des UAV abgeglichen wird und somit Hindernisse erkennen kann (Carloni et al., 2013). Jedoch ist die bildverarbeitungs-basierte Technik im Vergleich zum Lidar-System laut Merz und Kendoul (2013) im Hinblick auf die Genauigkeit und Robustheit weniger aussagekräftig. Es werden auch schon bekannte Techniken wie zum Beispiel das Radar von Flugzeugen eingesetzt. Hierbei werden Radarsensoren an dem UAV angebracht, die zur Hinderniserkennung beitragen (Kwag et al., 2006). Einer der bekanntesten UAV Hersteller „DJI“ nutzt für seine UAVs die Technik der omnidirektionalen Sicht. Dies ist ein Sensor, der die Fluglage des UAV schätzen kann (Mondragón et al., 2010). DJI verwendet zudem die Technik des Radars und kombiniert diese beiden Techniken zu einem omnidirektionalen digitalen Radar. Amazon hingegen benutzt bei seinen UAVs eine Kombination aus thermalen Kameras, Tiefenkameras und Sonar zur Erkennung von Hindernissen. Damit soll das UAV so stabil und robust wie ein herkömmliches Linienflugzeug fliegen können (Vincent & Gartenberg, 2019).

Insgesamt kann anhand der genannten Limitierungen erkannt werden, dass der Einsatz von UAVs für die nahe Zukunft fest geplant ist, jedoch mit vielen Herausforderungen und externen Restriktionen umgehen muss. Sei es das Problem der Batterieleistung und der damit verbundenen begrenzten Flugreichweite oder die noch nicht klar definierten gesetzlichen Restriktionen die zeigen, dass noch viele Probleme gelöst werden müssen bis die Paketauslieferung realisiert werden kann. Zudem wird die Sicherheitstechnik von UAVs noch weiterentwickelt werden müssen, bis UAVs die Erlaubnis bekommen voll autonom fliegen zu dürfen. In dem nächsten Kapitel werden die theoretischen Grundlagen zur Verdeutlichung der Simulation und des TSPs dargelegt.

### 3. Theoretische Grundlagen der Simulation und des Traveling-Salesman-Problems

In diesem Kapitel werden, abgegrenzt von den logistischen Grundlagen, die theoretischen Grundlagen, die für die folgende Simulation wichtig sind, beschrieben. Diese werden aufgeteilt in die Grundlagen einer Simulation und des TSPs. Innerhalb der TSP-Grundlagen wird sich explizit auf die TSPs für UAVs bezogen, da eines dieser TSPs für die Simulation verwendet wird.

#### 3.1 Grundlagen der Simulation und des Modells

Die Simulation wird im Bereich der Produktion und Logistik laut VDI wie folgt definiert: „Die Simulation ist das Nachbilden eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind“ (VDI, 2014, S. 3). Nach Law (2015, S. 1) wird die Simulation eingesetzt „[...] to imitate, or simulate, the operations of various kinds of real-world facilities or processes.“. Diese Anlagen oder Prozesse werden als Systeme bezeichnet. Die DIN IEC Norm 60050-351 beschreibt ein System als eine Menge miteinander in Beziehung stehender Elemente, die in einem bestimmten Zusammenhang als Ganzes gesehen und als von ihrer Umgebung abgegrenzt betrachtet werden (VDI, 2014). Demnach bilden die Anlagen und Prozesse nach Law (2015) eine bestimmte Menge miteinander in Beziehung stehender Elemente ab. Dadurch, dass die Paketauslieferung von UAVs noch nicht am realen System durchgeführt werden kann, wird die Auslieferung von Paketen durch UAVs als System betrachtet auf das eine Simulation angewendet wird, um Rückschlüsse auf die Realität zu bekommen.

Die Annahmen eines Systems, die in einer mathematischen oder logischen Beziehung stehen, bilden ein Modell (Law, 2015). Dieses Modell wird dann dazu genutzt, um das System zu verstehen (Law, 2015). Eine vergleichbare Definition von einem Modell liefern Gutenschwager et al. (2017) in der es heißt: „Zu Untersuchungszwecken werden in der Regel Modelle als Abbildung eines Systems erstellt. Auch zur Verdeutlichung bestimmter Sachverhalte wird auf die Bildung von Modellen zurückgegriffen.“ (Gutenschwager et al., 2017, S. 13). Diese beiden Definitionen decken sich zudem mit der des VDIs. Gutenschwager et al. (2017) folgern aus der Definition des VDIs zudem noch, dass ein Modell wiederum als System verstanden werden kann. Systeme oder Modelle können in statische und dynamische Systeme und Modelle unterteilt werden. Dynamische Systeme oder Modelle sind abhängig von der Zeit, welche durch die Systemzustände und die Zustandsänderungen beschrieben werden kann (Gutenschwager et al., 2017). Wobei bei statischen Systemen und Modellen die Ausgangsgröße zeitgleich mit der Eingangsgröße entsteht und damit Zeit unabhängig sind. Dynamische Systeme oder Modelle können zusätzlich in die Kategorie kontinuierlich und diskret in Bezug auf Zeit und Zustand eingeordnet werden (Gutenschwager et al., 2017).

Simulationsmethoden definieren, laut Gutenschwager et al. (2017), inwiefern das Zeitverhalten in der Simulation berücksichtigt wird. Daher werden die Simulationsmethoden in kontinuierlich und diskret unterteilt. Zusätzlich kann die diskrete Simulationsmethode in ereignisgesteuert und zeitgesteuert aufgeteilt werden. Durch diese Abgrenzung lassen sich zwei diskrete Simulationsmethoden ableiten, die

diskrete ereignisgesteuerte Simulationsmethode und die diskrete zeitgesteuerte Simulationsmethode. In der diskreten ereignisgesteuerten Simulationsmethode, auch ereignisdiskrete Simulation genannt, werden die Zustandsänderungen durch Ereignisse verursacht, die zu jedem Zeitpunkt eintreten können (Gutenschwager et al., 2017). Im Gegensatz dazu schreitet in der zeitgesteuerten Simulationsmethode die Zeit der Simulation in konstanten Zeitintervallen voran. Die Zustandsänderungen, die durch Ereignisse in einem zeitlichen Intervall entstehen, werden so erst am Ende des jeweiligen Intervalls verarbeitet. Ein Beispiel, das Gutenschwager et al. (2017) für die zeitgesteuerte Simulationsmethode nennt, ist ein mathematisches Modell, welches aus Gleichungen und Variablen besteht. Für das Aufstellen eines TSPs wird genau solch ein mathematisches Modell genutzt, um das TSP bestmöglich zu beschreiben. Demnach kann davon ausgegangen werden, dass das mFSTSP, welches in der Simulation dieser Arbeit angewendet wird, genau diese Simulationsmethode benutzt.

In Abbildung 1 sind die verschiedenen Wege zu erkennen, nach denen ein System erforscht werden kann. Zunächst stellt sich die Frage, ob das System bereits existiert oder es sinnvoll ist ein Experiment am realen System durchzuführen. Oft hat ein Experiment hohe Kosten zur Folge und würde das System zu sehr beeinflussen. Daher fällt die Wahl häufiger auf ein Modell, da durch ein Modell das reale System nicht beeinflusst wird und zunächst das System am Modell getestet werden kann (Law, 2015). Solch ein Modell kann entweder physisch oder mathematisch durchgeführt werden. Ein mathematisches Modell kann im Rahmen der Untersuchungen beliebig aufgebaut und manipuliert werden, um die Reaktion des Systems zu sehen. Daher ist ein mathematisches System laut Law (2015) dem physischen Modell, vorzuziehen.

Zum Schluss ist noch zu entscheiden, ob ein analytisches Ergebnis oder eine Simulation zur Auswertung des mathematischen Modells genutzt wird. Diese Entscheidung hängt laut Law (2015) von der Komplexität des mathematischen Modells ab. Sind das mathematische Modell und das System komplex und schwierig zu lösen, wird eine Simulation angewendet. Ein Simulationsexperiment kann beschrieben werden durch gezielte empirische Untersuchungen des Modellverhaltens auf Basis von wiederholten Simulationsläufen. Ein Simulationsmodell kann jedoch systematisch hinsichtlich seiner Parameter oder

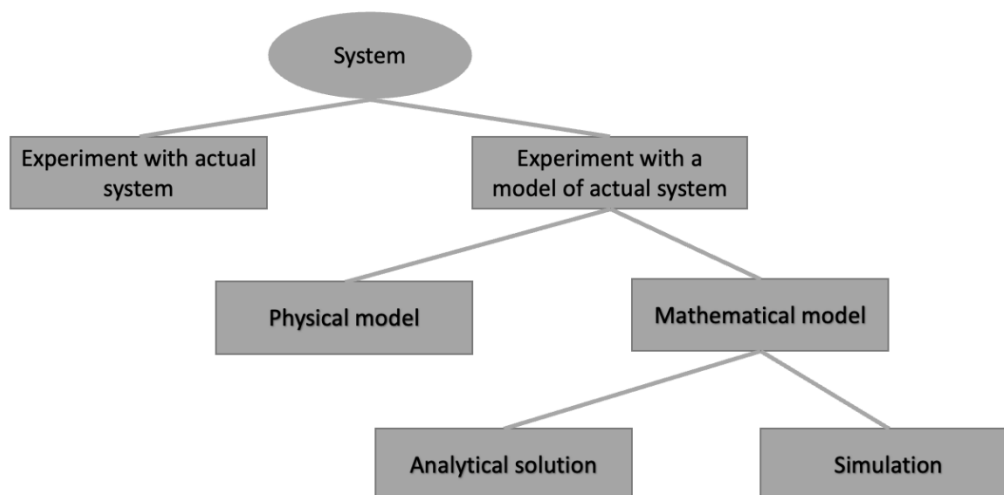


Abbildung 1: Wege um ein System zu Studieren in Anlehnung an Law (2015, S.4)

seiner Struktur variiert werden. Ein Simulationslauf wird als Ausführung des Simulationsmodells mit festgelegten Parametern verstanden (Gutenschwager et al., 2017).

Eine Definition zur Simulationswürdigkeit schildern Wenzel et al. (2008): „Eine Aufgabenstellung ist immer simulationswürdig, wenn die Lösung eines Problems nur mit der Methode Simulation gefunden werden kann. [...] Die Simulationswürdigkeit ist ebenfalls gegeben, wenn die Lösung mit anderen mathematischen Verfahren zwar möglich wäre, ein Simulationsmodell die Lösung aber wesentlich erleichtert.“ (Wenzel et al., 2008, S.15). Jedoch berücksichtigt diese Definition nicht die wirtschaftliche Perspektive.

Die Kosten einer Simulation können im Vorhinein gut kalkuliert werden. Es ist aber bis zum Ende der Simulation nicht möglich zu erkennen, wie hoch der Nutzen für das reale System sein wird. In Abbildung 2, in der zwei Kostenverläufe über die Zeit zu erkennen sind, werden die Kosten mit und ohne Simulation gegenübergestellt. Es fällt auf, dass die Kosten für die Planung und Realisierung mit Simulation höher sind als ohne Simulation. Dies ändert sich bereits in der Inbetriebnahme, da hier die Kosten ohne Simulation höher sind als die mit Simulation. Die höheren Kosten der Simulation in der Planungs- und Realisierungsphase werden in Kauf genommen, da dadurch spätere Fehler frühzeitig erkannt werden können. Somit entstehen in der Phase der Inbetriebnahme geringere Kosten für Fehlerausbesserungen. Zusätzlich kann die Kurve mit Simulation im Vergleich zu der Kurve ohne Simulation eine Zeitersparnis vorweisen. Die Simulation eines Systems ist genau dann wirtschaftlich und durchzuführen, wenn die entstehenden Kostenunterschiede zur Planung und Realisierung zwischen den beiden Kurven (Abbildung 2) mit und ohne Simulation kleiner ist als die Kostenunterschiede bei der Inbetriebnahme. Damit sind die Kosten der Simulation insgesamt kleiner als die ohne Simulation. Die Schwierigkeit beim Planen einer Simulation besteht also darin, die Kosten für die Fehlerausbesserungen und gleichzeitig die Höhe der Einsparung abzuschätzen wenn Simulation angewendet oder nicht angewendet wird (Gutenschwager et al., 2017).

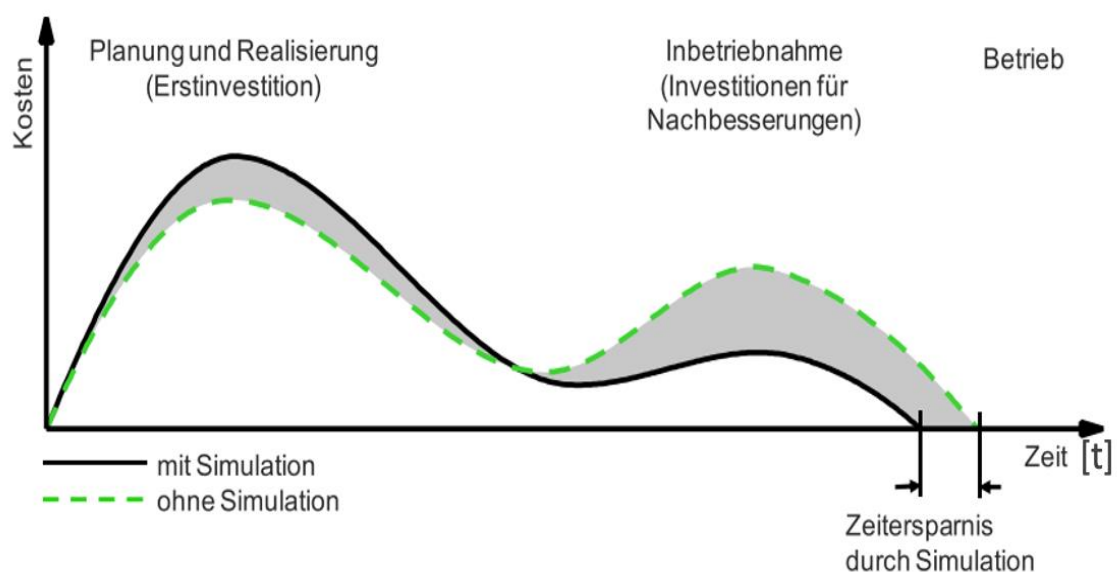


Abbildung 2: Schematischer Verlauf der Kosten von der Planung bis zur Inbetriebnahme in Anlehnung an Gutenschwager et al. (2017, S.48)



Das Vorgehen einer Simulationsstudie ist je nach Vorgehensmodell unterschiedlich. Eine Simulationsstudie wird in der Produktion und Logistik dann eingesetzt, wenn eine Simulation verwendet wird (Gutenschwager et al., 2017). Es gibt verschiedene Vorgehensmodelle für eine Simulationsstudie wie zum Beispiel von Maurice Landry et al. (1983), Banks (2014), Law und Kelton (2000) sowie von Rabe et al. (2008). Alle Vorgehensmodelle haben Gemeinsamkeiten aber auch Unterschiede in der Komplexität und Ausführung (Gutenschwager et al., 2017). In jedem Modell wird die Datenbeschaffung, das Konzeptmodell sowie die Implementierung thematisiert (Banks, 2014; Law & Kelton, 2000; Maurice Landry et al., 1983; Rabe et al., 2008). Das Vorgehensmodell nach Rabe et al. (2008) findet in der VDI Richtlinie 3633 (2014) Anwendung. Dies zeigt, dass das Modell von Rabe für die Produktion und Logistik in Deutschland, ein Standard für das Vorgehen von Simulationsstudien ist. Aufgrund dessen wurde sich in dieser Arbeit auf das Vorgehensmodell von Rabe et al. (2008) bezogen und keines der anderen Modelle weiter betrachtet. Im nachfolgenden wird das Modell weiter erläutert.

Das genannte Vorgehensmodell von Rabe et al. (2008) ist in Abbildung 3 zu sehen. Es ist zunächst zu erkennen, dass die einzelnen Phasen durch Ellipsen dargestellt werden. Die Phasen werden aufgeteilt in „Aufgabendefinition“, „Systemanalyse“, „Datenbeschaffung“, „Modellformalisierung“, „Datenaufbereitung“, „Implementierung“ sowie „Experimente und Analyse“ (Gutenschwager et al., 2017). Jeder genannten Phase wird wiederum ein Phasenergebnis zugeordnet, welches durch ein Rechteck dargestellt wird. Die Verbindung von Phase und Phasenergebnis wird durch einen Pfeil, der auf das Phasenergebnis gerichtet ist, ausgedrückt. Jedes Phasenergebnis ist Bestandteil eines größeren Rechteckes mit der Beschriftung „V & V der Daten & Modelle“ welches die Verifikation und Validierung ausdrückt. Eine

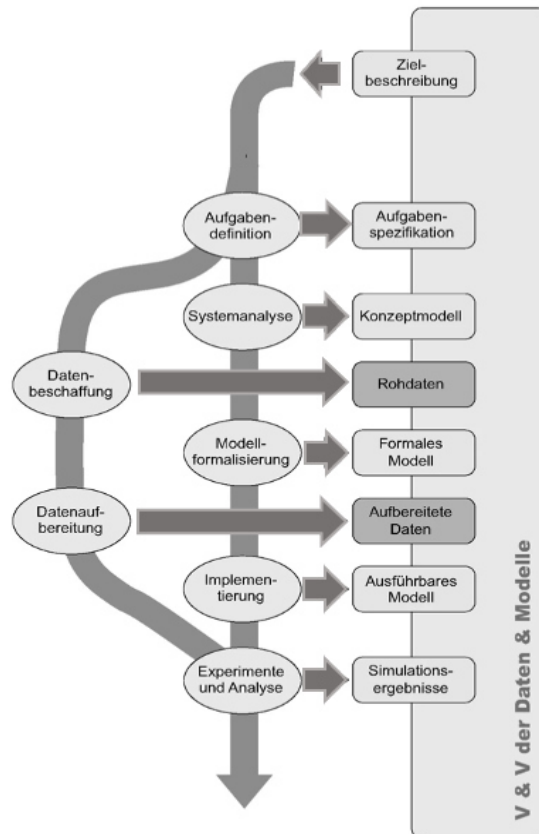


Abbildung 3: Vorgehensmodell bei der Simulation mit V&V in Anlehnung an Rabe et al. (2018, S.5)

nähere Erläuterung zur Verifikation und Validierung erfolgt zu einem späteren Zeitpunkt in diesem Kapitel. Innerhalb des Phasenablaufs gibt es zwei parallel verlaufende Pfade. Nach der Aufgabendefinition teilt sich der Pfad und kommt in der „Experimente und Analyse“-Phase wieder zusammen. Ein Pfad bildet sich aus „Systemanalyse“, „Modellformalisierung“ und „Implementierung“ mit den Phaseergebnissen „Konzeptmodell“, „Formales Modell“ und „Ausführbares Modell“. Diese drei Phasen lassen sich zusammen als Modellbildung konkretisieren (Gutenschwager et al., 2017). Der andere Pfad leitet sich aus „Datenbeschaffung“ und „Datenaufbereitung“ und den Phaseergebnissen „Rohdaten“ und „Aufbereitete Daten“ her. Diese Aufteilung entsteht dadurch, dass die Phasen „Datenbeschaffung“ und „Datenaufbereitung“ inhaltlich, zeitlich sowie bezüglich der einzubindenden Personen unabhängig von den anderen Phasen erfolgen kann (Gutenschwager et al., 2017). Bevor die sieben Phasen, aus Abbildung 3, erarbeitet werden, ist laut Rabe et al. (2008) eine „Zielbeschreibung“ anzufertigen. In dieser wird die Ausgangssituation, die Problemstellung und der Untersuchungszweck beschrieben. Des Weiteren werden der Projektumfang mit der Benennung von wesentlichen Elementen, die grobe Funktionsweise wesentlicher Ziele sowie erwartete Ergebnisaussagen und geplante Modellnutzung erläutert.

Die verschiedenen Phasen und deren Phaseergebnisse lassen sich wie folgt beschreiben:

- **Aufgabendefinition:** Konkretisiert die Inhalte der Zielbeschreibung in Bezug auf die vereinbarte Aufgabenstellung (Gutenschwager et al., 2017).  
Das Ergebnis der Phase, die Aufgabenspezifikation, wird als eine förmliche Übereinkunft der Projektpartner zu den geplanten Aufgaben und Zielen verstanden. Den Schwerpunkt der Aufgabenspezifikation bilden Angaben zum zu erstellenden Modell sowie des geplanten Nutzens. Des Weiteren wird festgelegt, welche Systemeigenschaften zu untersuchen sind und welche Experimente geplant werden. Eine beispielhafte Gliederung für die Aufgabendefinition ist in Rabe et al. (2008) auf der Seite 209 zu finden, sowie eine Checkliste für die Durchführung der Aufgabendefinition auf der Seite 199.
- **Systemanalyse:** In dieser Phase werden Festlegungen hinsichtlich des Umfangs und des Detaillierungsgrades des Modells getroffen (Gutenschwager et al., 2017).  
In dem Konzeptmodell, dem Ergebnis der Phase, werden Elemente des in der Zielbeschreibung benannten realen oder geplanten Systems, welche zu modellieren sind, genannt. Zudem werden die Genauigkeit, in der die Modellierung dieser Elemente erfolgen soll, und die Beziehung zwischen den Elementen festgehalten. Um dies zu erreichen werden vom VDI (2014) die systematischen Vorgehensweisen des Top-down- und Bottom-up-Entwurfs herangezogen (Gutenschwager et al., 2017).
- **Modellformalisierung:** Diese Phase sollte eine möglichst eindeutige Modellbeschreibung liefern, damit diese Schritte ohne eine weitere Erläuterung im Vorgehensmodell genutzt werden können (Gutenschwager et al., 2017).  
Das Ergebnis der Phase, das Konzeptmodell, präzisiert somit alle nicht formalen im Konzeptmodell enthaltenen Beschreibungen.
- **Implementierung:** Setzt das formale Modell mit einem Simulationswerkzeug in ein ablauffähiges Modell um. Diese Umsetzung macht meistens ein Wechsel der Beschreibungsmittel erforderlich, da die Logiken des formalen Modells in Programmiersprache umgewandelt werden. Parallel müssen alle noch nicht verständlichen Aspekte, welche sich aus dem ausführbaren Modell ergeben, dokumentiert werden (Gutenschwager et al., 2017).

- **Datenbeschaffung:** Dies geschieht parallel zu den vorher definierten Phasen der Modellbildung. In dieser Phase werden Informationen, die für die Simulationsstudie erforderlich sind, in Form von Rohdaten zusammengestellt. Informationen werden durch Daten repräsentiert, welche zweckgebunden sind und eine kontextbezogene Bedeutung (Semantik) aufweisen wobei Daten aus Zeichen mit Regeln ihrer Nutzung (Syntax) bestehen (Gutenschwager et al., 2017). Laut Gutenschwager et al. (2017) entsprechen alle Schritte von der Erhebungsplanung bis zur Datenerfassung der Datenbeschaffung.
- **Datenaufbereitung:** In dieser Phase werden die zuvor gesammelten Rohdaten so lange aufbereitet, bis sie für das ausführbare Modell nutzbar sind. Dies geschieht durch Simulationsfachleute, welche gleichzeitig die Eignung der Daten für die Aufgabenstellung validieren. Zudem werden alle Daten durch die genannten Fachleute vereinheitlicht, um eine gleichbleibende Granularität der Daten zu erlangen.
- **Experimente und Analyse:** Repräsentiert die Zusammenführung des formalen Modells und der aufbereiteten Daten. Die Experimente erfolgen auf Basis von Experimentplänen. Die Vorgehensweise der Experimente durch das statische und dynamische Faktor-Design lassen sich in Gutenschwager et al. (2017) auf den Seiten 178 bis 182 nachlesen. Die Simulationsergebnisse der Experimente werden im Anschluss mit Bezug auf die Zielbeschreibung ausgewertet und beinhalten zudem die Schlussfolgerung, wie das Modell nach Abschluss der Auswertungen zu bewerten ist.

Durch die Verifikation und Validierung der Daten und Modelle, werden die Phasenergebnisse auf mögliche Fehler oder Unzulänglichkeiten von Anfang an parallel zur Simulationsstudie geprüft damit entstandene Fehler nicht durch weitere Phasen fortgeführt werden. Zudem soll durch die Verifikation und Validierung sichergestellt werden, dass die erhaltenen Ergebnisse für die Beantwortung der Fragestellung verwendet werden dürfen (Gutenschwager et al., 2017). Die Verifikation ist laut Rabe et al. (2008) so zu verstehen: „Verifikation ist die Überprüfung, ob ein Modell von einer Beschreibungsart in eine andere Beschreibungsart korrekt transformiert wurde.“ (Rabe et al., 2008, S. 14). Die Validierung hingegen wird wie folgt definiert: „Validierung ist die kontinuierliche Überprüfung, ob die Modelle das Verhalten des abgebildeten Systems hinreichend genau wiedergeben.“ (Rabe et al., 2008, S. 15). Die verschiedenen Phasen und die Verifikation und Validierung bilden das gesamte Vorgehensmodell von Rabe ab. Die relevanten Phasen für diese Arbeit sind die Datenbeschaffung, die Datenaufbereitung, die Implementierung und die Experimente und Analysen. Die Phasen Aufgabendefinition, Systemanalyse und Modellformalisierung werden durch das Modell von Murray und Raj (2020) indirekt abgebildet.

Im nächsten Abschnitt werden die Grundlagen eines TSP verdeutlicht um zu zeigen, dass dies ein Bestandteil der Simulation ist.

## 3.2 Grundlagen eines Travelling-Salesman-Problems

Das TSP wird als ein Modell des Operation Research sowie der Mathematik und Informatik verstanden. Heuristiken, lineare Programmierung und ganzzahlige lineare Optimierung, welche die Grundlagen eines erfolgreichen Optimierungsproblems darstellen, wurden erstmals 1954 von Dantzig, Fulkerson und

Johnson für das TSP entwickelt (Michael Jünger et al., 1995). Seitdem werden diese zur Lösung praktischer Optimierungsprobleme eingesetzt.

Ein TSP kann wie folgt beschrieben werden: „Find the shortest route (tour) for a salesman starting from a given city, visiting each of a specified group of cities, and then returning to the original point of departure.“(Dantzig et al., 1954, S. 393). Allgemeiner erklärt, sind die Städte als symmetrische Matrix  $D = (d_{ij})$  anzusehen (Dantzig et al., 1954). Diese Matrix besitzt  $n \times n$  Felder. Die Felder sind mit den Entfernungen, aller möglichen Kombinationen, von Stadt  $i$  zu Stadt  $j$  zu reisen gefüllt. Solch ein Optimierungsproblem steigt in der Komplexität je nachdem wie viele Städte betrachtet werden und ob es asymmetrisch oder symmetrisch betrachtet wird. Bei einem Beispiel von  $n$  Städten, bei den das asymmetrische TSP angewendet werden soll, sind es bereits  $(n - 1)!$  Möglichkeiten. Für ein symmetrisches TSP halbieren sich die Optionen auf  $(n - 1)!/2$  Lösungsmöglichkeiten (Dantzig et al., 1954).

Das TSP wird durch ein mathematisches Modell, in dem die Nebenbedingungen beschrieben sind, entsprechend der Eigenschaften des zu untersuchenden TSPs eingeschränkt (Gutenschwager et al., 2017). Ein Beispiel für eine Nebenbedingung bei der Routenplanung ist laut Gutenschwager et al. (2017), das Festlegen eines Startpunkts oder Endpunkts. Ziel des mathematischen Modells ist es, die lineare Zielfunktion je nach Problem zu minimieren oder maximieren. So ordnet die Zielfunktion jeder möglichen Parameterkonfiguration der Entscheidungsvariablen einen festen Wert zu.

Es werden drei verschiedene Möglichkeiten zur Programmierung des mathematischen Modells in Gutenschwager et al. (2017) unterschieden. Zunächst gibt es die Lineare Programmierung (LP) bei der alle Modellvariablen und Entscheidungsvariablen reelle Zahlen aufweisen. Eine weitere Möglichkeit ist die Gemischt-ganzzahlige Programmierung (MILP), bei der eine Teilmenge der Modellvariablen eine Ganzzahligkeitsbedingung erfüllen muss. Der bekannteste Ansatz zur Lösung eines solchen Problems ist das Branch-and-Bound-Verfahren, für weiterführende Informationen wird auf Laporte und Nobert (1983) verwiesen. Zum Beispiel ist es für einen Autohändler, der ein Optimierungsproblem hinsichtlich der zu verkaufenden Autos hat, ratsam eine ganzzahlige Optimierung zu wählen, da dieser keine halben Autos verkaufen kann. Unterdessen deutet ein Optimierungsproblem für den Bauer, für die Menge an Dünger die er in Kilogramm braucht, auf ein lineares Optimierungsproblem hin. Als letzte Möglichkeit ist die Quadratische Programmierung (QP) zu nennen. Dieses Verfahren kann neben linearen Termen auch quadratische Ausdrücke enthalten (Gutenschwager et al., 2017).

Im Jahre 1972 wurde durch Korp das TSP als eines der ersten Probleme für NP-vollständig erklärt. Ein nichtdeterministisches in Polynomialzeit vollständiges Problem (NP-vollständig) ist nicht mehr effizient zu lösen und es wird gegebenenfalls eine suboptimale Lösung gewählt (Michael Jünger et al., 1995). Aufgrund der NP-Vollständigkeit des TSPs werden vermehrt Heuristiken auf das Problem angewandt. Laut Gutenschwager et al. (2017) können Heuristiken in Eröffnungsverfahren und Verbesserungsverfahren unterschieden werden. Das Eröffnungsverfahren baut an der Zielfunktion orientiert eine zulässige Lösung nach den vorgegebenen Nebenbedingungen auf. Diese werden genutzt, um eine Startlösung, für spätere Verbesserungsverfahren zu schaffen. Die Verbesserungsverfahren nutzen die Startlösungen, um diese sukzessiv zu verbessern (Gutenschwager et al., 2017). Typische Heuristiken, die für das TSP benutzt werden, sind Nearest-Neighbour, Insertion, gespannte Bäume und Saving-Methoden (Gutenschwager et al., 2017). Bei der Nearest-Neighbour Heuristik wird in einem Graphen immer der Nachbar besucht, der am nächsten vom aktuellen Knoten entfernt ist (Hurkens & Woeginger, 2004). Die Insertion Heuristik fängt zunächst mit einer kleineren Teilmenge von Knoten an und fügt nach und nach weitere Knoten hinzu. Es gibt verschiedene Verfahren, mit denen die Knoten eingefügt werden. Diese teilen

sich in nearest-, farthest-, cheapest- und random-Insertion auf (Michael Jünger et al., 1995). Die Saving-Heuristik vergleicht nach Michael Jünger et al. (1995) Subtouren miteinander, die von einem Basisknoten ausgehen. Es werden so lange Subtouren kombiniert, bis nicht mehr als eine Subtour existiert. Für jedes Paar von Subtouren wird die Ersparnis berechnet, die erzielt wird, wenn sie kombiniert werden. Nur die Subtouren der höchsten Ersparnisse werden kombiniert. Bei der Kombination werden die Kanten der einzelnen Subtouren zum Basisknoten gelöscht. Die entstandene Lücke wird durch die Kombination der Subtouren geschlossen (Michael Jünger et al., 1995).

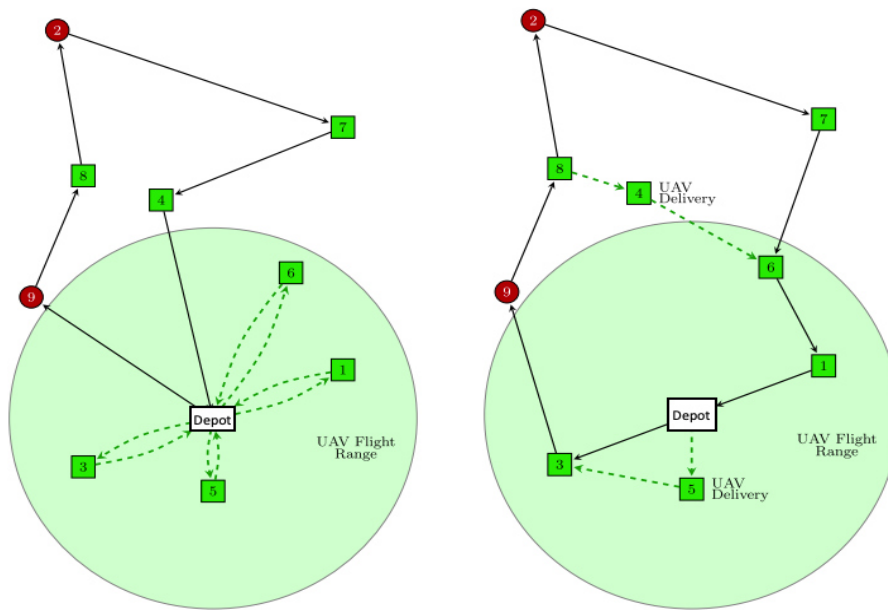
TSPs können für jegliche Routenoptimierungsprobleme aufgestellt werden. Eines der bekanntesten TSPs aus der Logistik ist die Touren-Planung. Hierbei wird versucht den vorhandenen Fuhrpark kostensparend, zeitminimierend und streckensparend einzusetzen (Lasch, 2020). Der nachfolgende Abschnitt behandelt TSPs die für die Paketauslieferung mittels UAVs aufgestellt worden sind.

### 3.3 Travelling-Salesman-Probleme für Unmanned-Aerial-Vehicles

Die Logistikunternehmen, welche die Zustellung mit UAVs planen, erhoffen sich durch die UAVs eine Reduzierung der Kosten, der benötigten Zeit eines Zustellprozesses und eine effizientere Gestaltung der letzten Meile (Dorling et al., 2017). Das TSP für UAVs berücksichtigt jedoch nicht nur die Routenoptimierung eines Handlungsreisenden, da in diesem Fall das Zustellfahrzeug und das UAV als jeweils eigene Handlungsreisende betrachtet werden. Im Allgemeinen soll das UAV, in naher Zukunft, autonom Pakete ausliefern. Jedoch muss das UAV vorher zugewiesen bekommen, welche Pakete, in welcher Reihenfolge, wohin geliefert werden müssen. Dieses Problem kann durch ein erweitertes TSP beschrieben werden. Die ersten, die ein solches TSP beschrieben haben sind Murray und Chu im Jahre 2015. Die beiden haben zwei Varianten der Zustellung mittels UAVs vorgestellt. Sie entwickelten das Flying-Sidekick-TSP (FSTSP), sowie das Parallel-Drone-Scheduling-TSP (PDSTSP). Ersteres bezieht sich auf die synchrone Auslieferung von einem UAV und einem Lastkraftwagen, wobei sich das PDSTSP auf die parallele Auslieferung bezieht (Murray & Chu, 2015). Beide TSPs haben das Ziel, die Zustellzeit zu minimieren.

In der Abbildung 4 (links) kann man die Vorgehensweise des PDSTSP genau erkennen. Die Pakete, die für das UAV zu schwer sind oder außerhalb des Zustellungsbereichs liegen, müssen vom Zustellfahrzeug zugestellt werden. Alle anderen Pakete kann das UAV selbst zustellen. Das UAV kehrt nach jeder Zustellung zum Depot zurück, da es je Zustellung nur ein Paket transportieren kann und wird neu beladen. Die Menge an Paketen, die der Paketbote mit dem Zustellfahrzeug zustellen muss, reduziert sich durch das PDSTSP. Das Vorgehen des FSTSP wird in Abbildung 4 (rechts) deutlich. Das UAV startet in diesem Beispiel vom Depot, fliegt zunächst den Kunden 5 an und kommt dann zum Zustellfahrzeug, welches inzwischen bei Kunde 3 angekommen ist. Das UAV ist nicht gezwungen vom Depot aus zu starten, jedoch ist es in diesem Beispiel durch das FSTSP so entschieden worden. So nimmt das UAV, im FSTSP, Zustellungen dem Zustellfahrzeug ab, die nicht auf dem Weg des Zustellfahrzeugs liegen und sich dadurch die Zeit des gesamten Zustellungsprozess verringert (Murray & Chu, 2015).

Diese beiden TSPs bildeten die Grundlage für die weitere Forschung an TSPs für UAVs. Es wurden beide Probleme in verschiedenen Bereichen weiterentwickelt. Das PDSTSP wurde von Dorling et al.



**Abbildung 4: Beispiel eines FSTSP und PDSTSP in Anlehnung an Murray und Chu (2015, S.88)**

(2017) mit einer Gemischt-ganzzahligen-Programmierung und mit dem Simulated-Annealing-Verfahren für den Fall, dass UAVs mehr als ein Paket gleichzeitig transportieren können, untersucht. Song et al. (2018) betrachten die Möglichkeit, dass UAVs Depots tauschen können, je nachdem ob ein anderes Depot besser für den Rückflug positioniert liegt oder nicht. Das TSP wurde auch, um die Möglichkeit von mehreren Depots und Trucks sowie ein Zeitfenster der Zustellung durch Ham (2018), erweitert. Das FSTSP wurde hinsichtlich der UAV-Flottengröße von Salama und Srinivas (2020) neu aufgestellt, indem sie zusätzliche Entscheidungsvariablen, zum Beispiel wie viele UAVs auf einen Truck sollten, berücksichtigen. Ein nahezu gleiches TSP wurde von Bouman et al. (2018) beschrieben. Dieses TSP nennt sich TSP-With-Drone (TSP-D). Das TSP-D beschreibt wie das FSTSP, eine synchrone Auslieferung mittels UAV und Zustellfahrzeug. Zusätzlich wurden verschiedene heuristische Lösungsansätze für das FSTSP eingeführt (Ha et al., 2020).

Murray erweiterte sein eigenes TSP im Jahre 2020 zu dem sogenannten mFSTSP. Dieses TSP berücksichtigt nun eine größere Anzahl von UAVs, die entweder vom Depot oder vom Zustellfahrzeug aus starten können. Die UAVs können in unterschiedlichen Modi Pakete ausliefern, die im Modell als *Vehicle-ID* implementiert sind. In diesen vier IDs, von 101-104, werden die Möglichkeiten zwischen einer weiten und kurzen Flugdistanz sowie zwischen einer niedrigen und hohen Geschwindigkeit abgedeckt. Das UAV erreicht somit laut Murray und Raj (2020) eine Geschwindigkeit von 15,6-31,3 m/s je nach eingestelltem Geschwindigkeitsmodus. Der Flugreichweitenradius kann je Modus zwischen 5 oder 10 Kilometern variieren. Bei diesen Angaben zur Flugreichweite und Fluggeschwindigkeit wurde sich auf die Parameter einer DJI Phantom 4 UAV gestützt. Weitere Annahmen die Murray und Raj (2020) einreichen, sind die Startzeit von 60 Sekunden für das UAV, Rüstzeiten für Batteriewechsel von 30 Sekunden, die Zustellzeit des Zustellfahrzeugs von 30 Sekunden und auch die Zustellzeit des UAV von 60 Sekunden. Ein weiteres Problem, welches bisher beim Einsatz von mehreren UAVs und einem Zustellfahrzeug auftrat, was das neue TSP nun berücksichtigt, ist das Problem der Warteschlange. Dies bedeutet, UAVs welche zum Zustellfahrzeug zurückkehren, während aber gleichzeitig noch ein anderes UAV

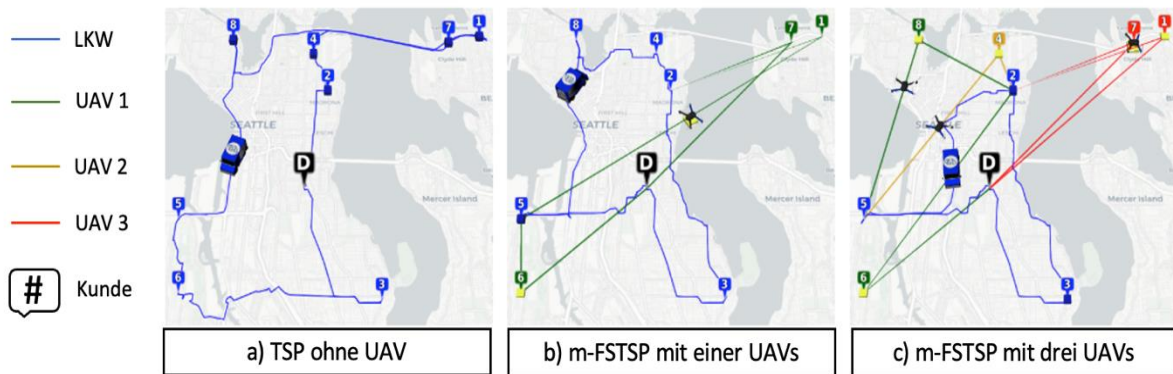


Abbildung 5: Beispiel des mFSTSP mit einem oder drei UAVs im Vergleich zur herkömmlichen Zustellung mittels Zustellfahrzeug in Anlehnung an Murray und Raj, 2020, S.3

im Zustellfahrzeug beladen wird, werden in eine Warteschlange geführt. Dieses Problem entsteht, laut Murray und Raj (2020), durch zu kleine Zustellfahrzeuge, die nicht mehr als ein UAV gleichzeitig an Board beladen können.

Zudem werden fünf verschiedene Ausdauer-Modelle der UAVs zur Spezifikation der Ausdauer und Batterieleistung von Murray und Raj (2020) betrachtet. Das erste Modell ist ein nicht lineares Modell, in dem der Energieverbrauch durch Paketgewicht und Fluggeschwindigkeit berechnet wird. Das zweite Modell ist ein lineares Modell, bei dem der Energieverbrauch nur durch das Paketgewicht bestimmt wird. Ein weiteres Modell, in dem alle UAVs unabhängig vom Paketgewicht und Fluggeschwindigkeit eine konstante Flugdauer zugewiesen bekommen, beruht auf einer festen, zeitbezogenen Flugdauer. Das vorletzte Modell berücksichtigt keine Parameter und nimmt die Ausdauer der UAVs als unbegrenzt an. Das letzte und fünfte Modell limitiert die Flugdauer aufgrund der Distanz zum Zustellort (Murray & Raj, 2020). Das für die Simulation relevante Ausdauer Modell ist jedoch das nichtlineare Modell, da nur in diesem Paketgewicht und Fluggeschwindigkeit berücksichtigt werden und dies die Aussage in Kapitel 2.3 zur Batterieleistung und -ausdauer stützt.

In Abbildung 5 sind die unterschiedlichen Fahrstrecken des Zustellfahrzeugs beim Einsatz von einem, zwei oder keinem UAV zu erkennen. Es ist ersichtlich, dass die Fahrstrecke des Zustellfahrzeugs abnimmt, wenn mehr UAVs das Zustellfahrzeug unterstützen. Welche Pakete die UAVs zustellen oder wann sie starten sollen, um optimal den Kunden zu erreichen, entscheidet das TSP und der Algorithmus dahinter. Entscheidungsvariablen können die Entscheidungen beeinflussen und werden daher in das mathematische Modell einbezogen. Das mFSTSP beruht auf der ganzzahligen linearen Optimierung, bei der die Variablen nur ganzzahlige Werte annehmen dürfen. Es existieren Entscheidungsvariablen für die UAV-Ausdauer, Vermeidung von Subtouren, UAV-Timing, Zustellfahrzeug-Timing sowie für Starts und Landungen (Murray & Raj, 2020).

Im Nachfolgenden werden als Beispiel die Zielfunktion und weitere Entscheidungsvariablen erläutert und alle weiteren Entscheidungsvariablen können in Murray und Raj (2020) nachgelesen werden.

$$\mathbf{Min} \quad \hat{t}_{c+1} \quad (1)$$

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} + \sum_{v \in V} \sum_{\substack{i \in N_0 \\ i \neq j}} \sum_{\substack{k \in N_+ \\ (v,i,j,k) \in P}} y_{vijk} = 1 \quad \forall j \in C \quad (2)$$

$$\sum_{j \in N_0} x_{0j} = 1 \quad (3)$$

$$\sum_{i \in N_0} x_{i,c+1} = 1 \quad (4)$$

$$\sum_{\substack{i \in N_0 \\ i \neq j}} x_{ij} = \sum_{\substack{k \in N_+ \\ k \neq j}} x_{jk} \quad \forall j \in C \quad (5)$$

$$\sum_{\substack{j \in C \\ j \neq i}} \sum_{\substack{k \in N_+ \\ (v,i,j,k) \in P}} y_{vijk} \leq 1 \quad \forall i \in N_0, \forall v \in V \quad (6)$$

Die Zielfunktion (1) versucht die Zeit, die vergeht bis das Zustellfahrzeug zum Depot zurückkehrt, zu minimieren. Alle weiteren Funktionen sind Entscheidungsvariablen, die die Zielfunktion beeinflussen oder Eingrenzungen für das Modell beinhalten.

Jeder Kunde muss genau einmal besucht werden, damit alle Pakete zugestellt werden können. Dies wird in Entscheidungsvariable (2) dargestellt. Dabei ist es egal ob der Kunde von dem UAV oder dem LKW besucht wird.

Damit das Zustellfahrzeug genau einmal vom Depot aus startet, wird die Entscheidungsvariable (3) aufgestellt.

Die darauffolgende Variable (4), lässt das Zustellfahrzeug genau einmal zum Depot zurückkehren, damit das Zustellfahrzeug am Depot den Zustellungsprozess beenden kann.

Zudem wird durch Variable (5) festgelegt, dass das Zustellfahrzeug von jedem Ort einmal losfahren muss, mit Ausnahme von dem Depot am Ende der Zustellung.

Die letzte genannte Variable (6) besagt, dass die UAVs von jedem Kunden zum LKW oder Depot zurückfliegen müssen. Ohne diese Variable könnte der Fall auftreten, dass die UAVs bei dem Kunden verweilen ohne zurückzukehren.

Da das TSP als NP-vollständig deklariert ist, wird für das Lösen einer größeren Menge von Kunden eine Heuristik benötigt. Diese Heuristik wird von Murray und Raj (2020) in drei Phasen aufgeteilt. In der ersten Phase werden die Kunden zu dem UAV oder dem LKW zugeordnet, je nachdem ob die Pakete den Limitierungen des UAV entsprechen oder nicht. Es wird zusätzlich eine mögliche LKW-Tour erzeugt. In der zweiten Phase werden die Einsätze für die UAVs generiert, die vorher in Phase I zugeordnet worden sind. Somit wird der Ort, an dem das UAV für den Kunden beladen wird, die Strecke, die das UAV zum Kunden fliegt, sowie wie und zu welchem Ort das UAV zum Zustellfahrzeug zurückkehrt,



festgelegt. In der dritten Phase wird, durch das Lösen eines MILP, der Zeitpunkt bestimmt, wann die einzelnen Aktivitäten stattfinden. Zudem wird die Warteschlangenreihenfolge ermittelt. Ist die Phase drei abgeschlossen werden die Phasen solange wiederholt, bis das TSP durch die Heuristik zu lösen ist (Murray & Raj, 2020).

Um das programmierte mFSTSP-Modell von Murray und Raj (2020) anwenden zu können werden zehn Argumente benötigt, die im Terminal auf dem Pfad des mFSTSP Ordners ausgeführt werden können. Der Funktionsaufruf des Modells ist wie folgt (Murray & Raj, 2019):

```
python main.py <problemName> <vehicleFileID> <cutoffTime> <problemType> <numUAVs>  
<numTrucks> <requireTruckAtDepot> <requireDriver> <Etype> <ITER>
```

Das Argument *problemName* ist ein Ordner, in dem die zwei für das Modell benötigten CSV-Dateien liegen. Die eine CSV-Datei beinhaltet alle Kunden sowie das Depot mit der Angabe von Längen- und Breitengraden sowie dem Paketgewicht in Pfund. Zusätzlich werden die Kunden mit einem *nodeType* von „1“ und das Depot mit „0“ beschrieben. Dies dient dazu, dass das Modell immer das Depot und die Kunden unterscheiden kann. Die zweite CSV-Datei beinhaltet jede mögliche Kombination von Routen vom Depot zu den Kunden und von einem Kunden zum anderen Kunden oder am Ende der Route zurück zum Depot (Murray & Raj, 2019). Dabei wird sich die Fahrstrecke in Metern und die für die Strecke beanspruchte Zeit in Sekunden angeschaut. Wie genau die Fahrstrecke und Zeit im Rahmen dieser Arbeit erhoben werden, kann in Kapitel 4 nachgelesen werden. Alle Lösungen, egal ob von der beschriebenen Heuristik oder vom Optimalen-Verfahren, werden in dem *problemName*- Ordner abgelegt. Die *vehicleFileID*, beschreibt wie die UAVs hinsichtlich Geschwindigkeit und Flugreichweite eingestellt sind. Hierbei werden die oben beschriebenen UAV-IDs 101-104 eingesetzt. Durch die *cutoffTime* wird dem Modell eine maximal erlaubte Laufzeit der Heuristik in Sekunden vorgegeben (Murray & Raj, 2019). Falls das Modell doch über die exakte MILP Programmierung und nicht über die Heuristik laufen soll, wird dies im *problemType* definiert. Für das Verwenden der Heuristik wird die „2“ und für die MILP Programmierung die „1“ als Wert angenommen. Die nächsten beiden Argumente sind *numUAVs* und *numTrucks*. Die Anzahl der UAVs, die zur Auslieferung eingesetzt werden sollen werden somit festgelegt. Die Anzahl von genutzten LKWs ist jedoch zurzeit noch auf einen LKW begrenzt, da der Einsatz von mehreren LKWs in diesem Modell nicht berücksichtigt wird. Daher ist hier der Wert auf „- 1“ gesetzt (Murray & Raj, 2019). Es ist möglich, dass der LKW nicht von Anfang an vom Depot aus starten muss. Dies kann durch das Argument *requireTruckAtDepot* festgelegt werden, zusätzlich können die UAVs nach Beendigung ihres Zustellprozesses zum Depot zurückfliegen ohne auf das Zustellfahrzeug zu warten. Das Argument *requireDriver* wird nur genutzt, wenn die UAVs nicht selbst vom LKW starten und landen können oder der Paketbote im Zustellfahrzeug anwesend sein muss. Das letzte Argument, was für das Modell benötigt wird, ist *ITER*. Dieses ist jedoch nur wichtig, wenn die Heuristik eingesetzt wird. *ITER* beschreibt die Häufigkeit, mit der die Heuristik durchlaufen werden soll. Bei der Verwendung der MILP-Programmierung wird *ITER* auf „-1“ gesetzt. Wurden alle Zehn Argumente berücksichtigt, korrekt eingegeben und die zwei erforderlichen CSV-Dateien im Problem-Ordner hinterlegt, kann das Modell ausgeführt werden (Murray & Raj, 2019). Das Modell generiert nach dem Ausführen des Modells eine Datei mit den Ergebnissen des Simulationslaufes und legt diese in dem Problem-Ordner ab.

Es ist ersichtlich, dass viele wissenschaftliche Arbeiten zu der Auslieferung von Paketen mittels UAVs in den letzten Jahren entstanden sind. Anfangs wurde bei den Modellen nur ein UAV und ein Zustellfahrzeug für die Auslieferung durch das FSTSP betrachtet. Nun kann in der Theorie durch das mFSTSP, mehr als ein UAV synchron mit dem Zustellfahrzeug die Pakete zustellen. Dies zeigt die schnell wachsende Forschungsentwicklung.

Zusätzlich werden die Batterieleistungen der UAVs berücksichtigt, da diese immer für die ganze Zustellung ausreichen müssen. Ferner können die einzelnen Argumente, die das Modell benötigt, in der Implementierung des Modells genutzt werden. Die Grundlagen der Simulation und des TSPs werden genutzt, um ein Modell aufzustellen, welches eine Zustellung von UAVs und Zustellfahrzeug in Dortmund abbilden kann. Weitere Annahmen aus den logistischen Grundlagen werden aufgegriffen, um das Modell an die Realität anzupassen. Aus diesen Annahmen werden verschiedene Simulationsläufe erstellt um das Modellverhalten gut beschreiben zu können. Die verschiedenen Simulationsläufe und die daraus gewonnenen Ergebnisse zeigen, ob das Ziel einer zeitlichen Minimierung des Zustellprozesses durch den Einsatz von UAVs zu realisieren ist.

Im nachfolgenden Kapitel wird die Zustellung mittels UAVs in Dortmund anhand des mFSTSP beschrieben. Dazu wird verdeutlicht wie das Modell von Murray und Raj (2020) verändert werden muss, damit dieses in Dortmund angewendet werden kann. Zusätzlich wird die Herangehensweise zur Implementierung des Modells in Dortmund genauer erläutert.

## 4. Theoretische Entwicklung der Paketauslieferung mittels Unmanned-Aerial-Vehicles und Lastkraftwagen in Dortmund

Der Markt für UAVs soll, laut der genannten Statistiken in Kapitel 2.2, einen starken Anstieg erfahren. Zudem kommt das steigende Interesse der Logistikunternehmen, die sich durch den Nutzen von UAVs eine effizientere Gestaltung der letzten Meile erhoffen. Dadurch, dass es noch kein reales System für die Paketauslieferung mittels UAVs existiert, wird die Paketauslieferung im Rahmen dieser Arbeit mithilfe einer Simulation betrachtet. Diese soll anschließend Rückschlüsse auf die Realität zulassen, um Fehler oder falsche Annahmen frühzeitig erkennen und lösen zu können.

In der theoretischen Entwicklung der Simulation zur Paketauslieferung mittels UAVs und LKWs in Dortmund ist es zunächst notwendig, geltende Annahmen und den Aufbau des Simulationsmodells aufzustellen. Damit diese das Simulationsmodell eingrenzen. Zudem wird sich auf die theoretischen Grundlagen bezogen indem bereits wichtige Bereiche der Simulation und bestehende Modelle zur Paketauslieferung mittels UAVs genannt wurden. Auf weitere Limitierungen und Restriktionen wird ebenfalls im Folgenden eingegangen. Alle erforderlichen Änderungen und Spezialisierungen des verwendeten Modells, damit dieses für die Stadt Dortmund anwendbar ist, werden in Kapitel 4.2 aufgeführt. Eine praktische Herleitung, wie die Daten für das Modell erhoben werden, findet sich im Kapitel 5.1.

### 4.1 Aufbau des Simulationsmodells

Das betrachtete Modell für die Simulation der Paketauslieferung mittels UAVs und LKWs in Dortmund, ist das mFSTSP von Murray und Raj welches erst im Jahre 2020 veröffentlicht wurde. Dieses Modell wurde aufgrund verschiedener Kriterien gewählt. Zum einem aufgrund der Aktualität und der Vollständigkeit. Wie in dem Kapitel 3.3 deutlich wurde, existieren zahlreiche Modelle zur Simulation von TSPs für UAVs. Die verschiedenen Schwerpunkte der anderen wissenschaftlichen Arbeiten können in dem oben genannten Kapitel nachgelesen werden. Das Modell von Murray und Raj kombiniert hingegen viele Theorien und Probleme in einem einzigen Modell. Wie zum Beispiel, die Verwendung von unterschiedlich vielen UAVs, die Betrachtung verschiedener Fluggeschwindigkeiten und Flugreichweiten. Eine genaue Lösung wird durch die MILP-Programmierung berechnet. Für eine größere Anzahl an Kunden wird eine Heuristik verwendet. Zum anderem wurden bestehende Probleme, wie das der Batterieleistung eines UAVs durch das Verwenden eines Ausdauermodells gelöst. Es wird erstmals von einem nichtlinearen Ausdauermodell ausgegangen. Dies bedeutet, dass die Batterieleistung und die Ausdauer des UAV nicht nur von einem Faktor, wie in den anderen genannten Ausdauermodellen, abhängig ist, sondern zusätzlich von dem Paketgewicht und der Geschwindigkeit des UAVs. Diese Annahme des nichtlinearen Ausdauermodells deckt sich zudem mit den getroffenen Aussagen zur Batterieleistung in dem Kapitel 2.3, da in den Grundlagen die Batterieleistung eines UAV auch von dem Paketgewicht und der Geschwindigkeit abhängig ist. Zusätzlich kann die Anzahl an verwendeten UAVs für bis zu vier UAVs angenommen werden. In dem verwendeten Modell wird nur ein Zustellfahrzeug, zur Paketauslieferung, betrachtet. Alle weiteren Annahmen, die in den Simulationsläufen getätigt werden, können in der Auflistung der geplanten Simulationsläufe nachgelesen werden.

Das Modell betrachtet, wie in Kapitel 3.1 erwähnt, eine zeitgesteuerte Simulationsmethode. Das mathematische Modell des TSPs wird so lange berechnet, wie es dem vorher eingegebenen, gewünschten Zeitintervall entspricht. Dadurch, dass das TSP als NP-Vollständig deklariert wird, ist das Modell und die optimale Lösung durch die MILP-Programmierung nur für eine geringe Kundenanzahl von maximal 8 Kunden technisch möglich. Der Grund hierfür ist, dass bei einer wachsenden Kundenanzahl die Berechnungsdauer exponentiell ansteigt. Für größere Kundenanzahlen wird daher die Heuristik des Modells verwendet.

Wichtige Annahmen, die für die Simulation getroffen werden sind zunächst, dass die Paketauslieferung durch UAVs und LKWs möglich ist. Limitierungen sowie Restriktionen der UAVs gelten als aufgehoben oder sind auf diese angepasst. Die Wetterbedingungen in der Simulation werden nicht berücksichtigt, sodass alle Simulationsläufe bei gleichen Bedingungen stattfinden. Ein UAV kann innerhalb der Simulation nur ein Paket gleichzeitig transportieren und hat, wie im Kapitel 2.2 beschrieben, eine Begrenzung des transportierbaren Paketgewichts von 2,5 Kilogramm. Es wird in allen Simulationsläufen vorausgesetzt, dass 75 Prozent der Pakete innerhalb des begrenzten Paketgewichts liegen. Die Flugreichweite und Geschwindigkeit werden in diesem Modell durch eine *Vehicle-ID* beschrieben. In den betrachteten Simulationsläufen wird sich auf die *Vehicle-ID 102* festgelegt, da diese die schnellstmögliche Zustellung mit den weitesten Distanzen, die für UAVs möglich ist, darstellt. Die technischen Angaben in dem Modell beziehen sich auf ein existierendes UAV von DJI, die Phantom 4, wie in Kapitel 3.3 nachgelesen werden kann. In dieser *ID* kann das UAV Kunden bis zu einem 10 Kilometerradius mit einer Fluggeschwindigkeit bis zu 31,3 *m/s* erreichen. Die angenommene Flugreichweite deckt sich zudem auch mit den Angaben aus Kapitel 2.2. Die maximale Flughöhe in der UAVs fliegen dürfen ist durch die EASA auf 120 Meter begrenzt. Jedoch in welcher Höhe UAVs Pakete ausliefern dürfen ist noch nicht rechtlich definiert, daher wird in der Simulation von einer Flughöhe von 50 Metern ausgegangen. Bei einer solchen Höhe entstehen weniger Hindernisse durch Häuser oder Bäume. Zusätzlich werden die Startzeit des UAV auf 60 Sekunden und die Rüstzeit, die beispielsweise durch ein Akkuwechsel entsteht, auf 30 Sekunden festgelegt. Die Dauer für eine Zustellung durch den Paketboten des Zustellfahrzeugs wird mit 30 Sekunden angenommen und die Zustellung durch das UAV mit 60 Sekunden. Zusätzlich wird für die Simulation entschieden die Argumente des Modells *requireTruckAtDepot* und *requireDriver* als *false* anzunehmen. So kann das UAV ohne den Truck zurück zum Depot fliegen, wenn es alle Zustellungen des Zustellprozesses getätigt hat. Der Paketbote profitiert auch davon, dass er nicht für jede Landung und jeden Start sowie zum Batteriewechsel im Zustellfahrzeug anwesend sein muss. Diese beiden Annahmen wurden getroffen, um das Ziel der Simulation einer minimalen Zustellzeit, weiter zu verstärken. Daher muss davon ausgegangen werden, dass es den UAVs bereits möglich ist automatisch zu Starten und zu Landen sowie einen automatischen Batteriewechsel durchzuführen.

Es werden verschiedene Simulationsläufe betrachtet, in denen sich die Parameter je nach Lauf ändern. Für die Simulation wird sich zum Einem auf einen innerstädtischen Stadtteil, mit einer unmittelbaren Nähe von weniger als 1,5 Kilometern zum Depot, und auf einen Vorort von Dortmund, der eine Entfernung von 8 Kilometern zum Depot besitzt, bezogen. Dies dient dazu, zu zeigen, wie sich das Modellverhalten bezüglich der unterschiedlichen Entfernung zum Depot auswirkt. Auch werden verschiedene Anzahlen von Kunden betrachtet. Die Simulationsläufe werden für 8, 20 und 50 Kunden erstellt. Diese Kundenanzahlen begründen sich zum einem, darauf das es für 8 Kunden möglich ist eine optimale Lösung durch die MILP Programmierung zu erhalten, und zum anderen um für höhere Kundenanzahlen

mittels der Heuristik das Modell in einer angemessenen Laufzeit ausführen zu können. Zusätzlich wird das Verhalten des Modells bei einer steigenden Kundenanzahl deutlich. Wird in dem Modell die MILP-Programmierung verwendet so muss für das Argument *problemType* eine 1 eingetragen werden, andernfalls eine 2 für die Heuristik. Jeder Simulationslauf wird mit keinem, einem oder drei UAVs berechnet. Der Fall, dass kein UAV zur Zustellung zur Verfügung steht wird betrachtet, um ein Vergleichswert für ein normales TSP zu erlangen. Daraus wird deutlich, wie stark die Anzahl von UAVs sich auf die benötigte Zeit des Zustellprozesses auswirkt. Somit wird das Argument *numUAVs* entweder auf 0, 1 oder 3 gesetzt. Es werden nicht mehr UAVs betrachtet, da das Modell nicht mehr berücksichtigt. Das Argument *numTrucks* wird wie bereits in Kapitel 3.3 beschrieben auf -1 gesetzt, da dem Modell die Betrachtung von nur einem Zustellfahrzeug möglich ist. Die Laufzeit für die MILP-Programmierung, oder auch *cutoffTime* genannt, wird in dieser Simulation auf 600 Sekunden begrenzt, da der benutzte Computer sonst die hohen Rechenleistungen nicht gewährleisten kann. Das Argument *problemName* liegt in einem Zeitstempelformat vor und muss in dem *Problems* Ordner des mFSTSP vorliegen, damit bei der Ausführung des Modells das Problem gefunden werden kann. In dem *problemName* Ordner befinden sich die zwei benötigten CSV-Dateien die vom Modell abgefragt werden. Wie bereits im Kapitel 3.3 erwähnt, betrachtet die Simulation das nichtlineare Ausdauermodell, welches im Argument *Etype* mit einer 1 zu deklarieren ist.

Im Nachfolgenden Abschnitt wird gezeigt, inwiefern das mFSTSP Modell modifiziert werden muss, damit es in Dortmund Anwendung finden kann.

## 4.2 Spezialisierung des Modells auf Dortmund

Das verwendete Modell mFSTSP von Murray und Raj aus dem Jahre 2020 wurde zunächst nur auf Städte wie Buffalo, New York, Seattle und Washington angewendet. Die Schritte, die es benötigt, um das Modell in Dortmund anwenden zu können, werden in diesem Abschnitt genauer erklärt. Zudem werden Besonderheiten, die in Dortmund gelten, dargelegt.

In der Simulation wird die Stadt Dortmund exemplarisch als Zustellort festgelegt. Diese Stadt weist, wie viele andere Ballungsgebiete, eine höhere Bevölkerungsdichte im innerstädtischen Bezirk auf als in den umliegenden Vororten. In diesen Vororten ist die Bebauungsstruktur aufgelockert und die Park sowie Verkehrsauslastung fällt weniger ins Gewicht. Zudem kann die Stadt ein zentral gelegenes Paketdepot im Gewerbepark Westfalenhütte aufweisen, dieses grenzt unmittelbar an die Innenstadt. Aufgrund der höheren Bevölkerungsdichte im innerstädtischen Bezirk kann somit zum einen die Paketauslieferung bei hoher Bevölkerungsdichte und nahem Bezug zum Depot, jedoch auch eine geringere Bevölkerungsdichte in einem Vorort von Dortmund, der weiter vom Depot entfernt ist, simuliert werden. Exemplarisch wird sich für den innerstädtische Bezirk der Stadtteil Dortmund-Ost und für den Vorort auf den Stadtteil Hombruch bezogen. Für eine Stadt wie Dortmund ist es wichtig das Verkehrsaufkommen im innerstädtischen Bezirk nicht durch Zustellfahrzeuge von Paketdiensten zusätzlich zu erhöhen. Positiv würde sich die Paketauslieferung mittels UAVs auf die Park- und Verkehrssituation in der Innenstadt auswirken, da nicht zusätzlich zu den PKWs noch mehrere Zustellfahrzeuge parken müssen. Durch den verringerten Verkehr in der Innenstadt kommt es zu einem geringeren CO<sub>2</sub>-Ausstoß. Zudem werden die benutzten UAVs zusätzlich elektrisch angetrieben und tragen so zu weniger CO<sub>2</sub> in der Innenstadt bei.

Das Modell von Murray und Raj muss daher so verändert werden, dass es Ergebnisse für Dortmund ausgeben kann. Dazu werden zwei CSV-Dateien erstellt. Die eine Datei fasst alle Kunden und das Depot in einer Tabelle zusammen. In dieser Tabelle wird eine Node-ID, Node-Type, Längengrad, Breitengrad, die Höhe in Metern über Normalnull und das Paketgewicht benötigt. In der anderen CSV-Datei befinden sich die Kosten in Sekunden sowie in Metern, die von Kunde zu Kunde und von Kunde zum Depot entstehen. Diese Kosten sind nicht zu verwechseln mit den Investitionskosten einer Simulation, wie in den Grundlagen Kapitel 3.1 genannt.

## **5. Simulation der Paketauslieferung mittels Unmanned-Aerial-Vehicles und Lastkraftwagen in Dortmund**

Dieses Kapitel beschäftigt sich mit der praktischen Anwendung der Paketauslieferung. Die genaue Implementierung der Daten für das Modell wird erklärt. Des Weiteren wird ein Programm beschrieben, in dem die benötigten CSV-Dateien für das Modell erstellt werden. Danach können die einzelnen Simulationsläufe durchgeführt und die resultierenden Ergebnisse dargelegt werden. Zum Abschluss werden die gewonnenen Erkenntnisse aus den Grundlagen und die erhobenen Ergebnisse der Simulationsläufe genutzt, um ein Fazit für die Paketauslieferung mittels UAV und LKW zu erstellen.

### **5.1 Exemplarische Implementierung des Modells**

Wie bereits im vorherigen Abschnitt erwähnt, werden für das Modell zwei CSV-Dateien benötigt. Diese müssen in einem bestimmten Format vorliegen, sodass das Modell diese verwenden kann. Im folgenden Abschnitt wird zuerst der Ablauf der Datenerfassung erklärt. Im Anschluss werden zwei Dateien, durch ein in Python geschriebenes Programm, generiert. Die Herangehensweise wird detailliert dargelegt, um eine einfache Anwendung dieser Datenerhebung für Dritte zu ermöglichen.

Der genaue Aufbau der beiden Dateien wurde in Kapitel 4.2 bereits erläutert und ist im Anhang auf Seite 39 tabellarisch dargestellt. Es wird ersichtlich, dass für die Dateien, geographische Daten erhoben werden müssen, durch die man einen genauen Längen- und Breitengrad eines beliebigen Standortes in Dortmund bestimmen kann. Zudem werden Routen generiert, um die Kosten des Zustellfahrzeugs, welche in Metern und Sekunden angegeben werden, darstellen zu können. Dafür werden alle möglichen Varianten von Routen, die von einem Kunden zum anderen Kunden verlaufen, benötigt. So entscheidet das Modell nach eigenen Berechnungen, wann welcher Kunde am effizientesten angefahren werden kann.

In der Abbildung 6 ist schematisch der Verlauf der Datenerhebung aufgezeigt. Jeder einzelne Schritt vom Beginn bis zum Erhalt der Ergebnisse aus dem Modell von Murray und Raj ist dargestellt. Vor der Datenerfassung müssen folgende Programme für die Anwendung installiert werden. Als erstes wird

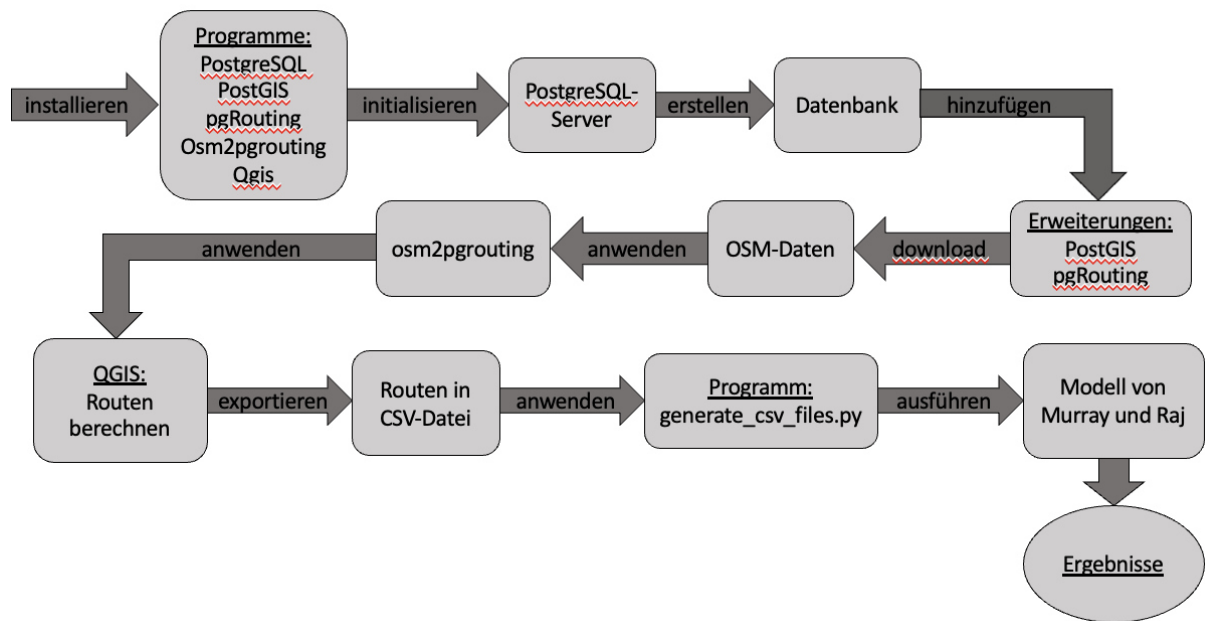


Abbildung 6: Schematischer Verlauf der Datenerhebung

*PostgreSQL*, oder auch kurz *Postgres*, benötigt. Es handelt sich dabei um ein objektrelationales Datenbankmanagementsystem (ORDBMS), welches ermöglicht eine Datenbank zu erstellen, in der die erhobenen Daten abgespeichert werden können. Geographische Daten können nicht direkt in die Datenbank aufgenommen werden, sie benötigen zunächst eine Erweiterung, die es möglich macht geographische Daten in der Datenbank abzulegen. Die besagte Erweiterung wird als *PostGIS* bezeichnet. Damit kann aus einer normalen Datenbank eine Geodatenbank erstellt werden, dessen Funktion es ist, geographische Daten abzuspeichern. Um Routen erstellen zu können, wird die zusätzliche Erweiterung *pgRouting* benötigt. Diese kann aus geographischen Daten Routen errechnen. Die Routen werden durch einen sogenannten Dijkstra-Algorithmus erstellt, der eine Route so berechnet, dass die geringsten Kosten entstehen. Das vorletzte Programm, welches Anwendung in der Datenerhebung findet, ist das *osm2pgrouting*. Es ermöglicht, geographische Daten aus Open-Street-Map (OSM) in eine Datenbank zu importieren, sodass diese importierten Daten von *pgRouting* verwendet werden können. Zum Schluss wird das Programm *QGIS* heruntergeladen, es visualisiert die OSM-Daten, hilft bei der Auswahl der Kundenstandorte und bei der Generierung der Routen. Nur wenn diese Programme alle fehlerfrei angewendet werden, kann die Erhebung der Daten beginnen.

Für diese Datenerhebung wird zunächst im Terminal des Computers ein Server initialisiert. Dies kann mit den zwei folgenden Befehlen in (7) durchgeführt werden.

```
initdb ~/my-postgres-server -E utf8
pg_ctl -D ~/my-postgres-server -l logfile start
```

(7)

Ist der Server von *Postgres* gestartet, kann die Datenbank für die Erhebung der geographischen Daten erstellt und die nötigen Erweiterungen hinzugefügt werden. Die Befehle für die Erstellung der Datenbank wird durch (8) ersichtlich.

```
createdb <name-db>
psql <name-db> -c 'CREATE EXTENSION postGIS'
psql <name-db> -c 'CREATE EXTENSION pgRouting'
```

(8)

Die erstellte Datenbank ist somit bereit die geographischen Daten von Dortmund aus OSM aufzunehmen. Die Verifikation und Validierung muss bei der Verwendung der Daten von OSM nicht eingreifen, da es sich um freie nutzbare Geodaten handelt und keine Lizenz erforderlich ist, sowie die Richtigkeit dieser angenommen werden kann. Zunächst muss die OSM-Datei aus OSM selbst heruntergeladen werden. Dies funktioniert bei OSM über den Exportbereich. Im Fenster des Exportbereichs muss die ganze Stadt Dortmund abgebildet werden, da sonst nicht alle Straßen, auf denen das Zustellfahrzeug später die Zustellungen durchführt, berücksichtigt werden und es zu einer fehlerhaften Routenberechnung kommen kann. Ist das notwendige Fenster ausgewählt, kann über den „Overpass API“-Button von OSM der Download der OSM-Datei gestartet werden. Für den nächsten Schritt wird *osm2pgrouting* verwendet. Wie in (9) zusehen, benötigt dieses Programm mehrere Informationen, um gestartet werden zu können. Zunächst wird die zuvor heruntergeladene OSM-Datei und der dazugehörige Dateipfad eingegeben. Das Programm benötigt zusätzlich eine *mapconfig.xml* Datei, welche auf der *osm2pgrouting* Github Internetseite heruntergeladen werden kann. Diese XML-Datei ist wichtig, um die relevanten Daten für *pgRouting* mit den OSM-Daten auszutauschen. Auch diese Datei wird mit dem gesamten Dateipfad angegeben.

```
osm2pgrouting --file<osm-datei> --conf<mapconfig-datei> --dbname <dbname> (9)
--username <username>
```

Um das Programm laufen zu lassen wird nach den zwei Dateien nur noch der Name der Datenbank und der Username zu dieser Datenbank benötigt. Konnte *osm2pgrouting* ohne Fehlermeldung ausgeführt werden, wurde die Datenbank mit den benötigten Daten gefüllt und kann nun visualisiert werden. Zu dieser Visualisierung wird das Programm *QGIS* verwendet. *QGIS* bietet die Möglichkeit, eine PostGIS-Datenbank mit dem Programm zu verbinden. Ist die Datenbank mit *QGIS* verbunden fehlt nur noch die Installation der *pgrouting* Erweiterung in *QGIS*. Nun kann das Straßennetz von Dortmund über *QGIS* dargestellt werden. Mit der *pgrouting* Erweiterung können nun Kunden sowie das Depot auf dem Straßennetz markiert werden. Ist die gewünschte Anzahl an Kunden markiert, können die Routen berechnet und die daraus entstandene Attributtabelle in eine CSV-Datei exportiert werden. Die Abbildung 7 zeigt ein Beispiel, wie die Visualisierung des Straßennetzes von Dortmund in *QGIS* aussieht. Außerdem kann

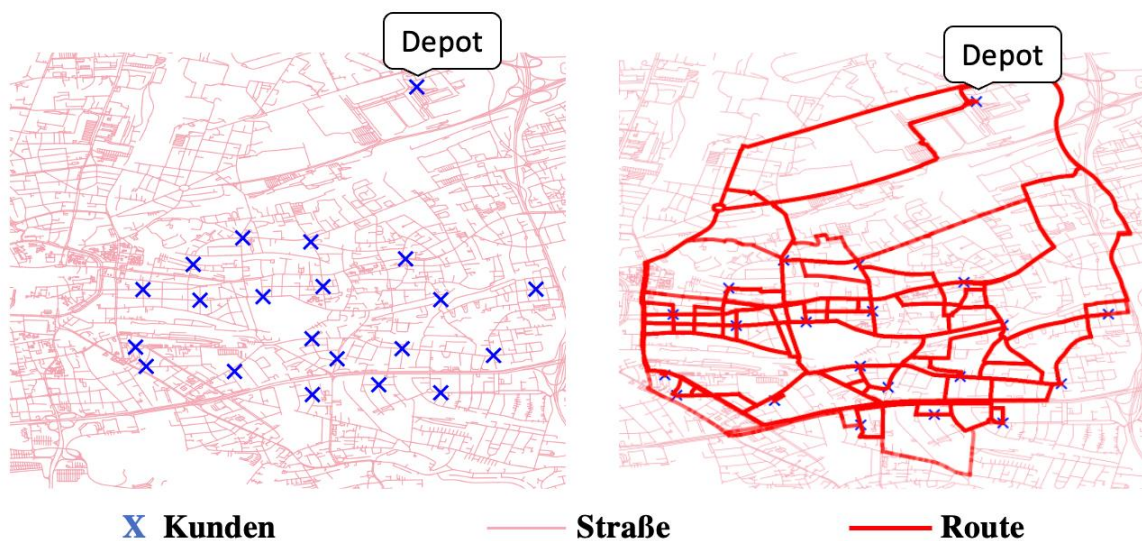


Abbildung 7: Beispiel einer Visualisierung in QGIS von 20 Kunden und der Routen



man die ausgewählten Kunden und das Depot, als Kreuze dargestellt, aus der Abbildung entnehmen. Auf der rechten Seite der Abbildung erkennt man den Verlauf der Routen im Einzelnen. In der Exportierten CSV-Datei sind nun alle relevanten Informationen, die für das Erstellen der zwei CSV-Dateien des Modells benötigt werden, enthalten.

Jetzt kann das in Python geschriebene Programm, mit dieser exportierten Datei aus *QGIS*, angewendet werden, um die beiden benötigten CSV-Dateien von Dortmund für das Modell zu generieren. In der besagten Datei liegen nun alle relevanten Daten der Routen, jedoch noch nicht in der richtigen Form, vor. Zu diesem Zeitpunkt sieht die Attributtabelle wie im Anhang in Tabelle 5 dargestellt aus. Alle Routen werden untereinander aufgelistet, allerdings mit sämtlichen Zwischenschritten, die zum Erreichen des Zielortes nötig sind. Eine Route mit ihren Zwischenschritten sind an den Spalten *\_path\_seq*, *\_start\_vid* und *\_end\_vid* zu erkennen. Die beiden Spalten *\_start\_vid* und *\_end\_vid* geben den Startpunkt und den finalen Endpunkt an. Die Zwischenschritte sind durch die Spalten *source* und *target* ersichtlich. Wie viele Zwischenschritte eine Route aufweist kann an der aufsteigenden Spalte *\_path\_seq* abgelesen werden. Diese nimmt erst wieder den Wert „1“ an, wenn eine neue Route mit neuen *\_start\_vid* und *\_end\_vid* betrachtet wird. Jede Spalte besitzt Informationen über die zeitlichen Kosten, die Länge in

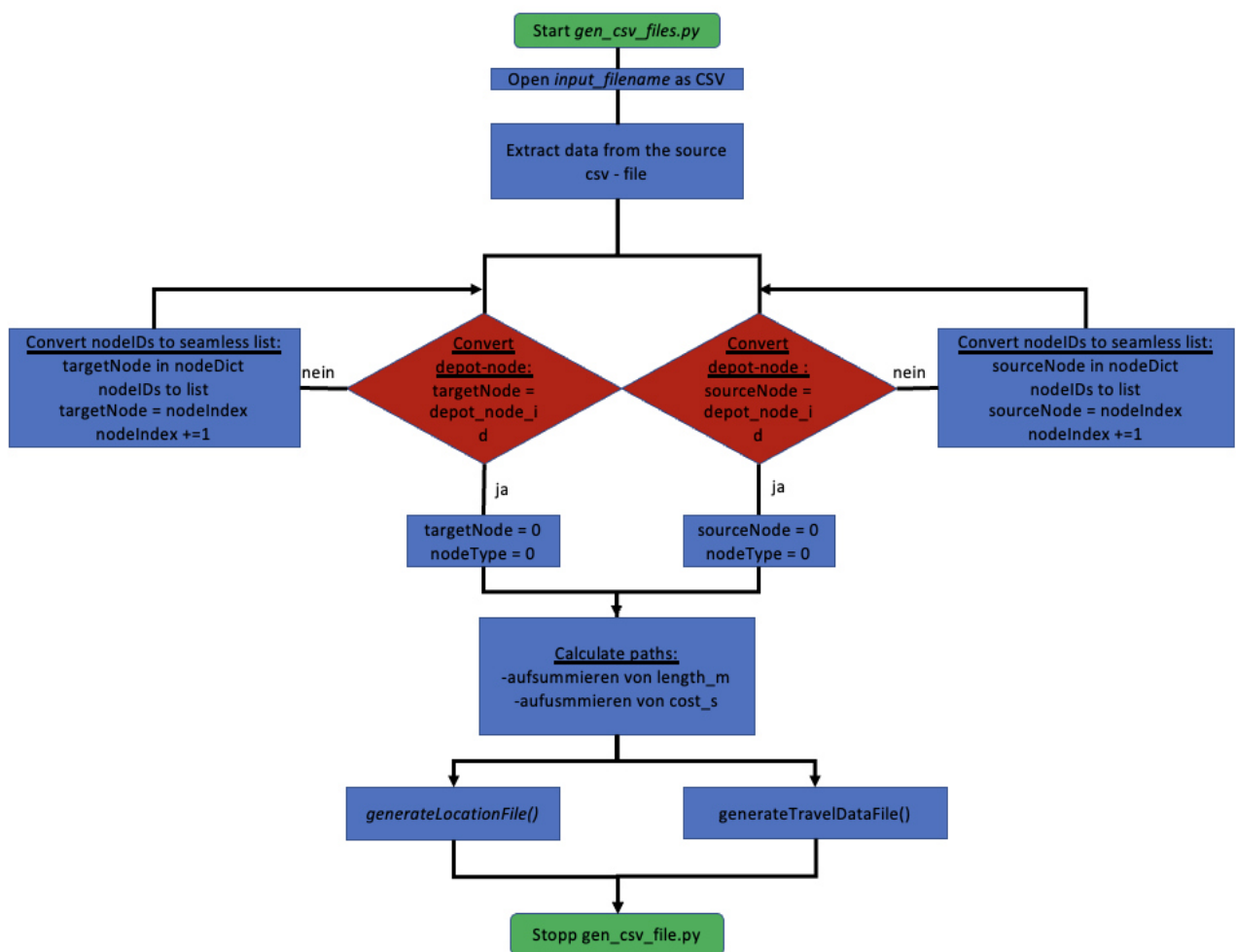


Abbildung 8: Exemplarischer Programmablaufplan von *gen\_csv\_files.py*

Metern der einzelnen Schritte und die Koordinaten dieser. Damit ist der Ort der relevanten Daten in der Attributtabelle bekannt jedoch liegen diese noch nicht in der relevanten Form vor.

Das Programm *gen\_csv\_files.py* zur Generierung der beiden CSV-Dateien wurde geschrieben, um diese Daten in die gewünschte Form zu bringen. In Abbildung 8 wird ein vereinfachter exemplarischer Programmablaufplan des Programms dargestellt. Zunächst ist der Name sowie der äußere Aufbau der Dateien wichtig, da sonst das Modell diese nicht erkennt. Das bedeutet, dass die Dateien am Ende mit „tbl\_locations.csv“ und „tbl\_truck\_travel\_data\_PG.csv“ benannt werden. Zudem müssen die Überschriften der einzelnen Spalten auch genau eine bestimmte Form aufweisen. In der *tbl\_truck\_travel\_data\_PG.csv* und in der *tbl\_locations.csv* muss die erste Überschrift mit einem Prozentzeichen beginnen, da sonst das Modell einen Fehler ausgibt. Die Überschriften für die *tbl\_truck\_travel\_data\_PG.csv* sind „% from location“, „to location“, „cost\_s“ und „length\_m“. Für die *tbl\_locations.csv* sind es „% nodeID“, „nodeType“, „latDeg“, „lonDeg“, „altMeters“ und „parcelWtLbs“. Ein weiteres äußeres Merkmal speziell für die *tbl\_locations.csv* ist es, dass das Depot immer die *nodeID* und *nodeType* von „0“ aufweisen muss. Zusätzlich steht das Depot immer in der ersten Zeile und alle weiteren Kunden darunter mit aufsteigender Nummer von „1“ angefangen. Die Kunden weisen jedoch den *nodeType* „1“ auf, um das Depot und die Kunden zu unterscheiden. Das Python-Programm füllt diese Tabelle anschließend mit den Koordinaten und der Höhenangaben der exportierten Datei aus *QGIS*. In dieser Datei fehlt noch das Paketgewicht. Dafür fügt das Programm eine zufällige Zahl zwischen eins und fünf in jede Spalte. In der Datei *tbl\_truck\_travel\_data\_PG.csv* wird auch das Depot mit „0“ deklariert und die Kunden mit einer aufsteigenden Zahl von eins beginnend. In dieser Datei werden die addierten Kosten jeder Route eingetragen. Dies bedeutet, dass die einzelnen Zwischenschritte bis zum Zielort in den Spalten *cost\_s* und *length\_m* summiert werden, somit können die gesamten Kosten in Metern und Sekunden abgebildet werden. Zum Schluss fehlt jedoch noch die Berücksichtigung der Pakete, die zu schwer für das UAV sind. In dem Kapitel 3.3 wurde angenommen, dass 75 Prozent der Pakete leicht genug sind, um mit UAVs ausgetragen zu werden. Daher müssen je nach Kundenanzahl 25 Prozent der Pakete auf ein Gewicht gesetzt werden, die das UAV nicht transportieren kann. Hierfür wird im Modell die Zahl „100“ verwendet, da diese das mögliche Gewicht eines UAVs übersteigt. Die CSV-Datei *tbl\_locations.csv* wird manuell in einem Tabellenbearbeitungsprogramm verändert. Sind alle Pakete die das Maximalgewicht überschreiten eingetragen, wird die CSV-Datei erneut exportiert und befindet sich nun in der endgültigen Form, die für das Modell erforderlich ist. Ein Beispiel der Input-Tabelle und der beiden Output-Tabellen sind im Anhang dargestellt (siehe Anhang S.39).

Werden die Dateien für das Modell wie beschrieben durch das Programm generiert, dann kann das Modell auf Dortmund angewendet werden. Die Durchführung der einzelnen Simulationsläufe und deren Ergebnisse werden im nächsten Abschnitt genauer betrachtet und ausgewertet.

## 5.2 Simulationsdurchführung und Erhalt der Ergebnisse

Das Konzept, wie es im vorherigen Kapitel aufgestellt wurde, kann nun zur Anwendung gelangen. Die Simulationsläufe werden, über den Funktionsaufruf wie in Kapitel 3.3 beschrieben, ausgeführt. Dabei wird eine Kundenanzahl von 8, 20 und 50 Kunden ausgewertet. Je Kundenanzahl wird die Zustellung

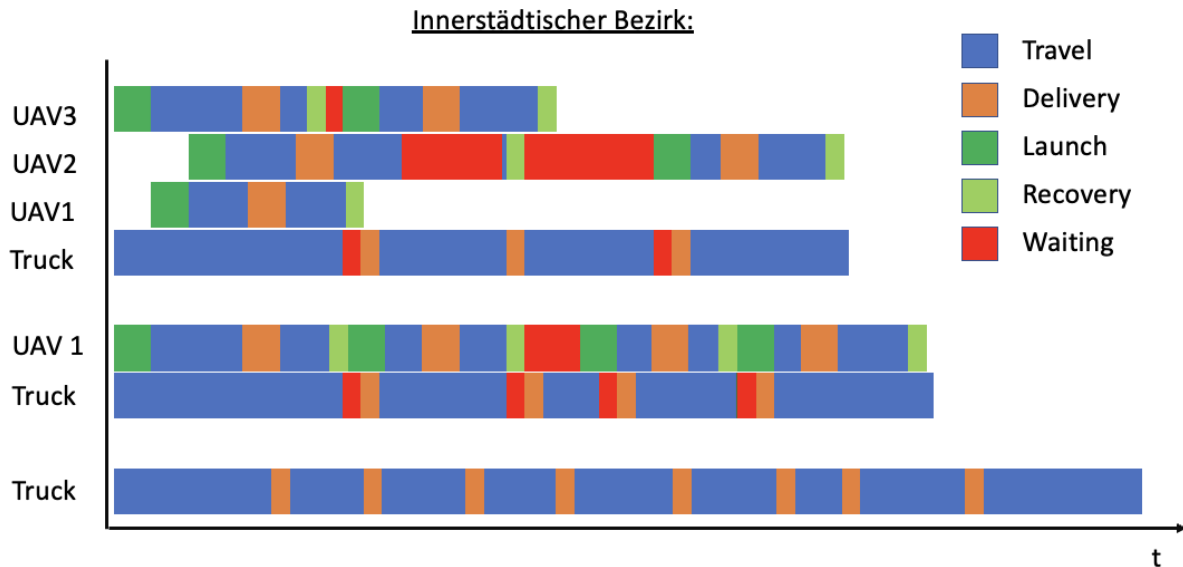


Abbildung 9: Gegenüberstellung der Auslieferungsvarianten im innerstädtischen Bezirk

mit keiner, einer oder drei UAVs, wie in Kapitel 4.1 begründet, betrachtet. Bei den Kundenanzahlen von 20 und 50 Kunden wird sich auf die Heuristik bezogen und nur bei 8 Kunden die exakte MILP-Programmierung betrachtet. In Kapitel 4.1 wird begründet, wann die Heuristik und die MILP-Programmierung verwendet wird. Die Argumente und Anzahl an Simulationsläufen, die in der Simulation betrachtet werden, sind in Tabelle 8 im Anhang dargestellt. Es entstehen somit je Zustellort und betrachteter Kundenanzahl, sowohl in der City als auch im Vorort, drei Ergebnis-Tabellen. Ein zusätzlicher Simulationslauf wird jeweils bei 8 Kunden in der City und im Vorort berechnet, um so die MILP-Programmierung und die Heuristik vergleichen zu können. In Summe werden 20 verschiedene Simulationsläufe betrachtet und ausgewertet. Aus diesen Tabellen der Simulationsläufe können Ersparnisse abgelesen und Auslieferungsvarianten im Hinblick auf die Zeit gegenübergestellt werden.

Zum einem wurden zwei Gantt-Diagramme, eins vom innerstädtischen Bezirk und das andere vom Vorort, erstellt in denen die benötigte Zeit der einzelnen Zustellvarianten visualisiert werden (siehe Abbildung 9 und Abbildung 10). In diesen werden die einzelnen Zeiten je Anzahl benutzter UAVs verglichen. Fünf verschiedene Zeiten werden im Diagramm aufgezeigt. Die *Travel*-Zeit beschreibt, wie lange das UAV oder der LKW fliegt beziehungsweise fährt. Der Zeitraum der die Zustellung des UAVs und

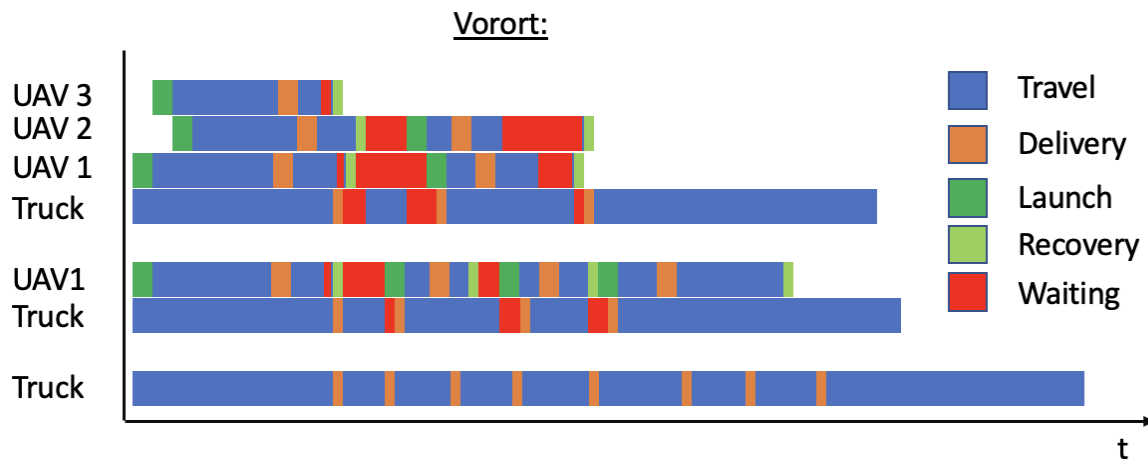


Abbildung 10: Gegenüberstellung der Auslieferungsvarianten im Vorort

LKWs darstellt, wird als *Delivery-Zeit* bezeichnet. Zwei weitere Zeiten werden speziell für das UAV betrachtet, nämlich die *Launch-Zeit* und die *Recovery-Zeit*. Die Bedeutung dieser beiden Zeiten kann in Kapitel 4.1 nachgelesen werden. Die letzte Zeit, die in dem Diagramm berücksichtigt wird, ist die *Waiting-Zeit*. Sie beschreibt, wie lange der LKW oder das UAV auf den jeweiligen anderen warten muss. Durch diese Betrachtungsweise kann man die verschiedenen Auslieferungsvarianten direkt untereinander vergleichen und die Ersparnisse werden ersichtlich.

Zum Anderem wurden die prozentualen Zeit-Ersparnisse je nach Kundengröße und Zustellort berechnet. Der Richtwert für die Berechnung der Ersparnis ist jeweils die Zeit, die nur das Zustellfahrzeug braucht, um die Pakete zuzustellen. Somit wird das Ziel dieser Arbeit, ob die Zustellung von Paketen mittels UAVs und Lastkraftwagen in Dortmund eine Zeitersparnis gegenüber der normalen Zustellung mittels Lastkraftwagen darstellt, berücksichtigt. In dieser Berechnung wird von dem Rückkehrzeitpunkt des LKWs im Depot, nach Beendigung des Zustellprozesses, ausgegangen. Die daraus resultierenden Ersparnisse sind in den drei Tabellen unten auf dieser Seite dargestellt. Zusätzlich neben der zeitlichen Ersparnis wurden die eingesparten Kilometer des Zustellfahrzeugs, die bei der Zustellung mittels UAVs entstehen, für 8 Kunden berechnet. Hierfür werden die Strecken der Zustellfahrzeuge ohne UAV und mit einer oder drei UAVs jeweils summiert. Daraus wird wiederum eine prozentuale Ersparnis berechnet. Es wird sich hierbei nur auf 8 Kunden bezogen, da die manuelle Auswertung der eingesparten Kilometer für eine höhere Kundenanzahl sehr zeitaufwändig ist und adäquate Ergebnisse wie mit 8 Kunden erwartet werden. Das Ergebnis dieser Berechnung ist Tabelle 4 zu entnehmen.

<u>8 Kunden:</u>		Innerstädtischer Bezirk		Vorort	
		[hr:min:sec]	% Savings	[hr:min:sec]	% Savings
TSP		00:27:31	-	00:47:47	-
mFSTSP	1	00:21:56	20.31%	00:38:35	19.26%
UAVs	3	00:19:41	28.47%	00:35:18	26.13%

Tabelle 1: Berechnete Zeit-Ersparnisse bei 8 Kunden

<u>20 Kunden:</u>		Innerstädtischer Bezirk		Vorort	
		[hr:min:sec]	% Savings	[hr:min:sec]	% Savings
TSP		00:46:43	-	01:03:19	-
mFSTSP	1	00:37:03	20.68%	00:56:34	10.68%
UAV	3	00:37:57	18.74%	00:53:41	15.21%

Tabelle 2: Berechnete Zeit-Ersparnisse bei 20 Kunden

<u>50 Kunden:</u>		Innerstädtischer Bezirk		Vorort	
		[hr:min:sec]	% Savings	[hr:min:sec]	% Savings
TSP		-	-	01:52:55	-
mFSTSP	1	01:08:57	-	01:27:20	22.66%
UAV	3	01:10:40	-	01:25:29	24.30%

Tabelle 3: Berechnete Zeit-Ersparnisse bei 50 Kunden

8 Kunden:		Innerstädtischer Bezirk		Vorort	
		Truck (km)	% Savings	Truck (km)	% Savings
TSP		15,86	-	32,11	-
mFSTSP	1	13,43	15.32%	26,13	18.62%
UAVs	3	13,19	16.83%	25,88	19.40%

Tabelle 4: Berechnete Einsparung der Kilometer für 8 Kunden

### 5.3 Auswertung der Ergebnisse

Aus den genannten Tabellen und Abbildungen ist ersichtlich, dass bei der Auslieferung mittels UAVs und Lastkraftwagen Zeitersparnisse entstehen. So kann für das Ziel dieser Arbeit festgestellt werden, dass das betrachtete Modell der Paketauslieferung in der Simulation schneller durchgeführt werden kann als die Zustellung mit Zustellfahrzeug.

Dies wird anhand der Gantt-Diagramme in Abbildung 9 und Abbildung 10 ersichtlich. Unbeachtet ob der innerstädtische Bezirk oder der Vorort betrachtet wird, der Balken der Zustellungsvariante des normalen TSP, im Diagramm „Truck“ genannt, ist länger als die Balken der Modellvarianten die aus dem mFSTSP entstehen. Eine genauere zeitliche Ersparnis wurde in den Tabellen 1 bis 3 auf der Seite 31 berechnet. Diese werden im Folgenden beschrieben. Bei der Betrachtung von 8 Kunden im innerstädtischen Bezirk kann die höchste Ersparnis, von 28,47 Prozent mittels Zustellfahrzeug und drei UAVs, erzielt werden. Eine etwas geringere Ersparnis von 26,13 Prozent weist die Zustellung von 8 Kunden mit drei UAVs und LKW im Vorort auf. Wie vorher beschrieben ist es für 8 Kunden möglich die exakte MILP-Programmierung zu berechnen, um ein Vergleich zu erlangen, wie sehr die Heuristik von der MILP-Programmierung abweichen kann. Die Heuristik berechnet eine Zustellzeit von 37 Minuten und 35 Sekunden. Die prozentuale Abweichung der Heuristik zur exakten MILP-Programmierung beträgt somit 5,66 Prozent.

Für die nachfolgenden Ergebnisse wird sich nur noch auf die Ergebnisse der Heuristik bezogen, da für 20 und 50 Kunden keine exakte MILP-Programmierung mit dem zur Verfügung stehenden Endgerät möglich ist. Werden 20 Kunden betrachtet verhält es sich wie folgt. Drei UAVs im innerstädtischen Bezirk brauchen länger zur Zustellung, als wenn das Zustellfahrzeug nur mit einer UAV die Pakete zustellt. Die Zeitersparnis mit drei UAVs beträgt nur 18,74 Prozent, wobei ein UAV eine Ersparnis von 20,68 Prozent aufweisen kann. Dieses Verhalten ist jedoch nicht im Vorort zu erkennen. Hier kann die Zustellung mit drei UAVs eine Ersparnis von 15,21 Prozent aufweisen, welche höher ist als bei einer Zustellung mit einer UAV. Wenn die Zustellung mittels Zustellfahrzeug und keiner UAV für 50 Kunden im innerstädtischen Bezirk berechnet wird, sind die resultierenden Ergebnisse nicht aussagekräftig. Das Modell gibt einen Funktionswert von „-1“ wieder und berechnet keine Route für das Zustellfahrzeug. Für alle weiteren Berechnungen bei der Betrachtung von 50 Kunden, im innerstädtischen Bezirk oder im Vorort, funktioniert das Modell wie gewünscht. Es wurden ebenfalls neue Daten aus OSM erhoben, um zu prüfen, ob der Fehler durch falsche Kartendaten entstanden ist. Trotz neuer Kartendaten trat dieser Fehler dennoch nur bei der Betrachtung von 50 Kunden und keiner UAV im innerstädtischen Bezirk auf. Es ist jedoch davon auszugehen, dass sich auch bei 50 Kunden im innerstädtischen Bezirk eine zeitliche Ersparnis einstellen wird. Ergebnisse für die Zustellung mit einer oder drei UAVs können vom

Modell ausgewertet werden. Die Berechnung der Ersparnis in Bezug auf die Zustellung nur mit Zustellfahrzeug entfällt dadurch. Es ist jedoch auch hier zu erkennen, dass die Zustellung mit drei UAVs länger dauert als mit nur einer UAV. Im Vorort können drei UAVs eine Ersparnis von 24,30 Prozent vorweisen und haben somit eine bessere Ersparnis als mit einer UAV im Vorort.

Es muss zusätzlich beachtet werden, dass die Daten von OSM die aktuelle Verkehrssituation in der Innenstadt und im Vorort nicht abbilden können. Vorhandene Baustellen, Staus oder stockender Verkehr und Unfälle fließen nicht in die Simulation ein. In Tabelle 4, werden die gefahrenen Kilometer des Zustellfahrzeugs im Vergleich zur Zustellung mittels einer oder drei UAVs betrachtet. In jedem betrachteten Beispiel werden Kilometer durch den Einsatz von UAVs eingespart. Die höchsten Einsparungen können jedoch im Vorort mit drei UAVs nachgewiesen werden. Dies ist darauf zurückzuführen, dass im Vorort die einzelnen Kunden weiter voneinander entfernt wohnen als im innerstädtischen Bezirk. Werden nun Kunden von UAVs im Vorort beliefert, können diese eine größere Distanz für das Zustellfahrzeug einsparen als wenn die Kunden näher zusammenwohnen würden, wie es in der Innenstadt der Fall ist. Im städtischen Bezirk ist der Unterschied der Einsparung in Kilometern zwischen der Auslieferung mit einer oder drei UAVs kaum gegeben. Das Zustellfahrzeug mit drei UAVs fährt 25 Meter weniger als das mit einem UAV. Wie bereits vorher an der zeitlichen Ersparnis zu erkennen ist, ist somit der Einsatz von drei UAVs in dem innerstädtischen Bezirk nicht empfehlenswert. Sie scheinen sich viel mehr gegenseitig zu behindern. Dies kann anhand der Wartezeiten im Gantt Diagramm in Abbildung 9 belegt werden. Für drei UAVs und Zustellfahrzeug beträgt die Wartezeit des gesamten Zustellprozesses 633 Sekunden. Im Gegensatz weist die Zustellung mit einer UAV und Zustellfahrzeug nur eine Wartezeit von 153 Sekunden auf.

Demnach ist bei einer höheren Bevölkerungsdichte und dem betrachteten Modell von Murray und Raj, der Einsatz von nur einer UAV und dem Zustellfahrzeug für das Unternehmen effizienter. Wird das PDSTSP-Modell wie in Kapitel 3.3 beschrieben betrachtet, werden alle Kunden, die die Voraussetzungen erfüllen und in Reichweite des UAVs wohnen, vom Depot aus mit einem UAV beliefert. Nur die Pakete, die nicht von der UAV zugestellt werden können oder nicht in der Reichweite liegen, werden vom Zustellfahrzeug ausgeliefert. Auf die in diesem Modell betrachtete Beispielstadt Dortmund würde das PDSTSP im innerstädtischen Bezirk eine gute Alternative darstellen. Das betrachtete Depot von Dortmund liegt innenstadtnah und durch die angenommene Reichweite des UAVs liegt der gesamte Innenstadt Bereich in Reichweite des UAVs. Bei geringerer Bevölkerungsdichte und größerer Entfernung zum Depot kann sich das mFSTSP eher durchsetzen. Durch weite Distanzen zwischen den einzelnen Kunden können die UAVs das Zustellfahrzeug besser unterstützen als in Ballungsgebieten wie Dortmund. Wirkungsvoller für das mFSTSP scheinen ländliche Umgebungen zu sein, welche nicht in Reichweite eines Depots liegen. So kann die Geschwindigkeit und Flugreichweite der UAV optimal ausgenutzt werden und zur zeitlichen Minimierung des Zustellprozesses beitragen und so Kosten in der Zustellung sparen.

Die weiteren angenommenen Annahmen wirken sich auch auf die zeitliche Ersparnis aus. Wie in Kapitel 4.1 angenommen, kann das UAV ohne Paketboten auf dem Zustellfahrzeug selbst landen, Pakete laden und den Batteriewechsel durchführen. Zusätzlich muss für eine Rückkehr des UAVs zum Depot nicht zwingend das Zustellfahrzeug im Depot anwesend sein. Würden diese beiden Annahmen nicht gelten bräuchte die Zustellung insgesamt länger, da das UAV und der Paketbote öfter aufeinander warten müssten. Eine andere Annahme, der maximalen Flugreichweite und Geschwindigkeit des UAVs, welche durch die *vehicleID 102* ausgedrückt wird, wirkt sich auch auf die zeitliche Ersparnis aus. In dieser

Annahme wird von der schnellsten Geschwindigkeit und der weitesten Flugreichweite, die im Modell betrachtet werden kann, ausgegangen. Wie aber schon in Kapitel 3.3 erklärt, sind diese Geschwindigkeiten und Reichweiten von bereits existierenden UAVs erreichbar und können somit als realistische Werte angenommen werden. Das Modell kann nur ein Zustellfahrzeug annehmen, daher fällt die Betrachtung von mehreren Zustellfahrzeugen in diesem Modell weg. Es können letztlich keine Aussagen getroffen werden, wie mehrere Zustellfahrzeuge sich auf das Modell auswirken würden. Das betrachtete Ausdauermodell, welches die Flugreichweite zusätzlich von Gewicht und Geschwindigkeit abhängig macht, wird in diesem Modell angenommen. Dadurch kann die Ausdauer des UAVs realistischer abgebildet werden, wie in Kapitel 2.3 erklärt.

Das Ziel von Logistikunternehmen, welches in Kapitel 2.1 bereits genannt wurde, ist die effizientere Gestaltung der Letzten-Meile. Werden die zeitlichen Ersparnisse dieses Modells betrachtet scheint dieses Modell sehr vielversprechend zu sein. Jedoch erkennt man schon in den innerstädtischen Bezirken, dass bei größerer Kundenanzahl der Einsatz von mehr als drei UAVs länger dauert, als wenn ein UAV eingesetzt wird. Daher ist zu entscheiden, ob in der Innenstadt sich eher auf das PDSTSP bezogen wird, und auf ländlicheren Zustellungsgebieten das hier betrachtete Modell angenommen wird.

## 5.4 Fazit

Durch das in dieser Arbeit erstellte Modell und den daraus erhaltenen Ergebnissen für die Paketauslieferung mittels UAV und Lastkraftwagen in Dortmund ist klar geworden, dass diese Art der Paketauslieferung der im Vergleich zur herkömmlichen Paketauslieferung mittels Zustellfahrzeug und Paketbote zeitlich gesehen vorzuziehen ist. Jedoch wurden durch die Ergebnisse des Modells auch die Schwächen desselben deutlich.

Es können nicht beliebig viele Lastkraftwagen und UAVs in dem Modell zur Auslieferung betrachtet werden und gibt dem Anwender somit weniger Freiheiten in der Simulation. Zudem hat das Modell gezeigt, dass es bei einer hohen Bevölkerungsdichte und dem Einsatz von mehreren UAVs mit Zustellfahrzeug länger zur Zustellung braucht, als wenn ein UAV und Zustellfahrzeug die Pakete ausliefert. Das verwendete Modell von Murray und Raj ist daher für eine geringere Bevölkerungsdichte und auf ländlicheren Gebieten, ohne Nähe zum Depot, anzuwenden. Für höhere Bevölkerungsdichten und kurzer Entfernung zum Depot kann das PDSTSP eine gute Alternative darstellen. In diesem kann zum Beispiel in Dortmund die gesamte Innenstadt als Zustellgebiet für UAVs abgedeckt werden. Das Zustellfahrzeug wird dabei nur für den Transport von nicht in Reichweite liegende Kunden und von Paketen, die das Maximalgewicht überschreiten, genutzt.

Wird Bezug auf die genannten Herausforderungen für eine effizientere Letzten-Meile in Kapitel 2.1 genommen, so werden diese durch das Modell zum Teil erfüllt. Nachhaltiger im Vergleich zur herkömmlichen Zustellmethode sind beide Modelle, da weniger Zustellfahrzeuge genutzt, gefahrene Kilometer des Zustellfahrzeugs eingespart werden und die UAVs ohne CO<sub>2</sub>-Ausstoß fliegen. Der Zeitdruck, den die Logistikunternehmen haben, wird durch eine schnellere Zulieferung geschmälert und somit können mehr Pakete insgesamt zugestellt werden. Die älter werdenden Arbeitnehmer werden durch die hohe Automatisierung des Modells von vielen Arbeitsschritten entlastet. Ob dies jedoch langfristig zu einer

komplett automatischen Zustellung entwickelt wird und so eine Gefahr der Arbeitslosigkeit eines Paketboten mit sich bringt, lässt sich zu diesem Forschungsstand dieser Zustellungsmethode nicht sagen.

Wie hoch sich die Kosten dieser Zustellungsmethode für die Logistikunternehmen belaufen, kann nicht aus dem Modell entnommen werden und bedarf weitere eigene Studien und Simulationen. Durch die Anwendung von Simulation erhoffen sich die Unternehmen, wie in Abbildung 2 in Kapitel 3.1 dargestellt, eine zeit- und kostensparende Inbetriebnahme des realen Modells. Obwohl durch die Simulation zunächst hohe Investitions- und Simulationskosten entstehen, amortisieren sich die Kosten durch die Anwendung dieser.

## 6. Zusammenfassung und Ausblick

Das Ziel dieser Arbeit, zu zeigen ob die Auslieferung mittels UAV und Lastkraftwagen im Vergleich zur herkömmlichen Zustellung mittels Zustellfahrzeug und Paketboten in Dortmund möglich gemacht werden kann und zeitliche Ersparnisse hervorbringt, wurde erreicht.

Trotzdem ist zu erwähnen, dass die erhaltenen Ergebnisse abhängig von der Größe des betrachteten Modells sind. Daher sind in diesem Fall keine anderen Auswertungen möglich als die in dem Modell angenommen werden können. Ein Beispiel dafür ist einerseits, dass nicht mehr als ein Zustellfahrzeug betrachtet werden kann und andererseits, dass nicht beliebig viele UAVs mit dem Zustellfahrzeug zusammenarbeiten können. Dadurch fehlt die realistischere Betrachtungsweise von mehr als einem Zustellfahrzeug welches Pakete ausliefern kann. Bei der Bewertung der Ergebnisse dieser Arbeit sind die getroffenen Annahmen zu berücksichtigen, die in dem erarbeiteten Konzept impliziert wurden. Eine dieser Annahmen ist, dass die UAV automatisch landen, sich beladen und den Batteriewechsel durchführen kann. Alle getroffenen Annahmen beeinträchtigen das Ergebnis des Modells auf eine andere Weise. Zusätzlich stellt sich die Frage, ob diese Annahmen auf die Wirklichkeit bezogen richtig angenommen wurden.

Für die weitere Entwicklung der Paketauslieferung mittels UAVs und Lastkraftwagen sollten sich zukünftige wissenschaftliche Arbeiten auf die nachfolgenden Punkte spezialisieren. Die Betrachtung des Luftraums, welcher eine große Rolle spielen wird, wenn tausende Pakete per UAV über den Luftraum zugestellt werden. Zusätzlich muss sich das betrachtete Modell von Murray und Raj weiterentwickeln und dem Anwender mehr Möglichkeiten bieten. Dies kann zunächst die Möglichkeit sein, mehrere Zustellfahrzeuge zu betrachten oder beliebig viele UAVs fliegen zu lassen. Weiterführend könnten sich mehrere Zustellfahrzeuge, die in dem Luftraum befindlichen UAVs eines Logistikunternehmens teilen, sodass nicht eine UAV zu einem Zustellfahrzeug gehört, sondern zu allen Zustellfahrzeugen zurückfliegen kann.

Das in dieser Arbeit dargestellte Vorgehen zur Erhebung der relevanten Karten-Daten von Dortmund aus OSM und dem Erstellen der benötigten CSV-Dateien für das Modell von Murray und Raj kann durch wenige Schritte auf andere Städte übertragen werden. Lediglich ein anderer Ausschnitt von OSM-Daten, beispielsweise der Ausschnitt von Berlin, muss über OSM heruntergeladen werden. Diese Daten können wie beschrieben durch *osm2pgrouting* in die Datenbank eingelesen werden und in QGIS visualisiert werden. Alle weiteren Schritte bis zum Ausführen des Modells bleiben, wie zuvor in Kapitel 5.1 beschrieben, gleich.



## Literaturverzeichnis

- Aurambout, J.-P., Gkoumas, K. & Ciuffo, B. (2019). Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities. *European Transport Research Review*, 11(1). <https://doi.org/10.1186/s12544-019-0368-2>
- Banks, J. (2014). *Discrete-event system simulation* (5. Aufl.). Always learning. Pearson.
- BMVI. (2021). *Zum Jahreswechsel gelten neue EU-Regelungen für Drohnen*. <https://www.bmvi.de/SharedDocs/DE/Artikel/LF/drohnen.html>
- Bouman, P., Agatz, N. & Schmidt, M. (2018). Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4), 528–542. <https://doi.org/10.1002/net.21864>
- Boysen, N., Fedtke, S. & Schwerdfeger, S. (2021). Last-mile delivery concepts: a survey from an operational research perspective. *OR Spectrum*, 43(1), 1–58. <https://doi.org/10.1007/s00291-020-00607-8>
- Carloni, R., Lippiello, V., D'Auria, M., Fumagalli, M., Mersha, A. Y., Stramigioli, S. & Siciliano, B. (2013). Robot Vision: Obstacle-Avoidance Techniques for Unmanned Aerial Vehicles. *IEEE Robotics & Automation Magazine*, 20(4), 22–31. <https://doi.org/10.1109/MRA.2013.2283632>
- Dantzig, G., Fulkerson, R. & Johnson, S. (1954). Solution of a Large-Scale Traveling-Salesman Problem. *Journal of the Operations Research Society of America*, 2(4), 393–410. <https://doi.org/10.1287/opre.2.4.393>
- de-Miguel-Molina, M. & Santamarina-Campos, V. (2018). *Ethics and Civil Drones*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-71087-7>
- Dorling, K., Heinrichs, J., Messier, G. G. & Magierowski, S. (2017). Vehicle Routing Problems for Drone Delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1), 70–85. <https://doi.org/10.1109/TSMC.2016.2582745>
- EASA. (2019). *Easy Access Rules for Unmanned Aircraft Systems (Regulations (EU) 2019/947 and (EU) 2019/945)*.
- Gruber, J., Kihm, A. & Lenz, B. (2014). A new vehicle for urban freight? An ex-ante evaluation of electric cargo bikes in courier services. *Research in Transportation Business & Management*, 11, 53–62. <https://doi.org/10.1016/j.rtbm.2014.03.004>
- Gutenschwager, K., Rabe, M., Spieckermann, S. & Wenzel, S. (2017). *Simulation in Produktion und Logistik*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-55745-7>
- Ha, Q. M., Deville, Y., Pham, Q. D. & Hà, M. H. (2020). A hybrid genetic algorithm for the traveling salesman problem with drone. *Journal of Heuristics*, 26(2), 219–247. <https://doi.org/10.1007/s10732-019-09431-y>
- Ham, A. M. (2018). Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, 91, 1–14. <https://doi.org/10.1016/j.trc.2018.03.025>
- Hänsenberger, S. (2018). *Die zivilrechtliche Haftung für autonome Drohnen unter Einbezug von Zulassungs- und Betriebsvorschriften*. Carl Grossmann. <https://doi.org/10.24921/2018.94115928>
- Heutger, M. & Kückelhaus, M. Dr. (2014). UNMANNED AERIAL VEHICLES IN LOGISTICS: A DHL perspective on implications and use cases for the logistics industry.

- Hurkens, C. A. & Woeginger, G. J. (2004). On the nearest neighbor rule for the traveling salesman problem. *Operations Research Letters*, 32(1), 1–4. [https://doi.org/10.1016/S0167-6377\(03\)00093-2](https://doi.org/10.1016/S0167-6377(03)00093-2)
- Kwag, Y. K., Choi, M. S., Jung, C. H. & Hwang, K. Y. (2006). Collision Avoidance Radar for UAV. In *2006 CIE International Conference on Radar* (S. 1–4). IEEE. <https://doi.org/10.1109/ICR.2006.343231>
- Laporte & Nobert (1983). A branch and bound algorithm for the capacitated vehicle routing problem.
- Lasch, R. (2020). *Strategisches und operatives Logistikmanagement: Distribution*. Springer Fachmedien Wiesbaden. <https://doi.org/10.1007/978-3-658-31869-7>
- Law, A. M. (2015). *Simulation modeling and analysis* (5. Aufl.). *McGraw-Hill series in industrial engineering and management science*. McGraw-Hill Education.
- Law, A. M. & Kelton, W. D. (2000). *Simulation modeling and analysis* (3. Aufl.). *McGraw-Hill series in industrial engineering and management science*. McGraw-Hill. <http://www.loc.gov/catdir/description/mh023/99052146.html>
- Maurice Landry, Jean-Louis Malouin & Muhittin Oral. (1983). *Model validation in operations research*. <https://doi.org/10.5194/acp-2016-545-RC1>
- Mensen, H. (2013). *Handbuch der Luftfahrt*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-642-34402-2>
- Merz, T. & Kendoul, F. (2013). Dependable Low-altitude Obstacle Avoidance for Robotic Helicopters Operating in Rural Areas. *Journal of Field Robotics*, 30(3), 439–471. <https://doi.org/10.1002/rob.21455>
- Michael Jünger, Gerhard Reinelt & Giovanni Rinaldi (1995). Travelling Salesman Problem.
- Mondragón, I. F., Campoy, P., Martinez, C. & Olivares, M. (2010). Omnidirectional vision applied to Unmanned Aerial Vehicles (UAVs) attitude and heading estimation. *Robotics and Autonomous Systems*, 58(6), 809–819. <https://doi.org/10.1016/j.robot.2010.02.012>
- Murray, C. C. & Chu, A. G. (2015). The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, 54, 86–109. <https://doi.org/10.1016/j.trc.2015.03.005>
- Murray, C. C. & Raj, R. (2019). *The Multiple Flying Sidekicks Traveling Salesman Problem (mFSTSP): Test problems and source code for solving the Multiple Flying Sidekicks Traveling Salesman Problem (mFSTSP)*. <https://github.com/optimotorlab/mFSTSP#the-multiple-flying-sidekicks-traveling-salesman-problem-mfstsp>
- Murray, C. C. & Raj, R. (2020). The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies*, 110, 368–398. <https://doi.org/10.1016/j.trc.2019.11.003>
- Otto, A., Agatz, N., Campbell, J., Golden, B. & Pesch, E. (2018). Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey. *Networks*, 72(4), 411–458. <https://doi.org/10.1002/net.21818>
- Rabe, M., Spieckermann, S. & Wenzel, S. (2008). *Verifikation und Validierung für die Simulation in Produktion und Logistik*. Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-540-35282-2>
- Rojas Viloria, D., Solano-Charris, E. L., Muñoz-Villamizar, A. & Montoya-Torres, J. R. (2021). Unmanned aerial vehicles/drones in vehicle routing problems: a literature review.

- International Transactions in Operational Research*, 28(4), 1626–1657.  
<https://doi.org/10.1111/itor.12783>
- Salama, M. & Srinivas, S. (2020). Joint optimization of customer location clustering and drone-based routing for last-mile deliveries. *Transportation Research Part C: Emerging Technologies*, 114, 620–642. <https://doi.org/10.1016/j.trc.2020.01.019>
- Song, B. D., Park, K. & Kim, J. (2018). Persistent UAV delivery logistics: MILP formulation and efficient heuristic. *Computers & Industrial Engineering*, 120, 418–428.  
<https://doi.org/10.1016/j.cie.2018.05.013>
- Statista 2021a. Prognose der Entwicklung der Nachfrage auf dem Drohnenmarkt in Deutschland bis 2025. <https://de.statista.com/statistik/daten/studie/972655/umfrage/prognose-der-entwicklung-der-nachfrage-auf-dem-drohnenmarkt-in-deutschland/>
- Statista 2021b. Umsatz durch E-Commerce in Deutschland bis 2019. <https://de.statista.com/statistik/daten/studie/3979/umfrage/e-commerce-umsatz-in-deutschland-seit-1999/>
- ten Hompel, M., Bauernhansl, T. & Vogel-Heuser, B. (2020). *Handbuch Industrie 4.0* (3. Auflage). Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-58530-6>
- VDI (2014). VDI-Richtlinie 3633 Blatt 1: Simulation von Logistik-, Materialfluss- und Produktionssystemen Grundlagen.
- Vincent, J. & Gartenberg, C. (6. Mai 2019). Here's Amazon's new transforming Prime Air delivery drone. *The Verge*. <https://www.theverge.com/2019/6/5/18654044/amazon-prime-air-delivery-drone-new-design-safety-transforming-flight-video>
- Wenzel, S., Weiß, M., Collisi-Böhmer, S., Pitsch, H. & Rose, O. (2008). *Qualitätskriterien für die Simulation in Produktion und Logistik*. Springer Berlin Heidelberg.  
<https://doi.org/10.1007/978-3-540-35276-1>

# Anhang

## Tabellen der Input- und Output-Dateien des *gen\_csv\_files.py* Programms

<u>CSV-Datei von QGIS:</u>		<b>INPUT</b>							
<u>_path_seq</u>	<u>_start_vid</u>	<u>_end_vid</u>	source	target	length_m	cost_s	x1	y1	
1	658	3220	658	4233	18,06	2,17	7.5124363	51.5057515	
2	658	3220	4233	4255	127,26	15,27	7.5121803	51.5057805	
3	658	3220	4255	4253	20,37	2,44	7.5120548	51.5046394	
...	...	...	...	...	...	...	...	...	
1	658	3507	658	4233	18,06	2,17	7.5124363	51.5057515	
2	658	3507	4233	15242	80,31	9,64	7.5121803	51.5057805	
...	...	...	...	...	...	...	...	...	

Tabelle 5: Beispiel der Input-Datei des *gen\_csv\_files.py* Programms

<i>tbl_truck_travel_data_PG.csv</i>			<b>1.OUTPUT</b>	
% from location	to location	cost_s	length_m	
0	0	0	0	
0	1	824,26	11975,71	
0	2	787,07	11756,26	
...	...	...	...	
1	0	1004,24	12905,59	
1	1	0	0	
1	2	101,39	903,71	
...	...	...	...	

Tabelle 6: Beispiel der ersten Output-Datei des *gen\_csv\_files.py* Programms

<i>tbl_locations.csv</i>		<b>2.OUTPUT</b>			
%nodeID	nodeType	latDeg	lonDeg	altMeters	parcel-WtLbs
0	0	7.5060037	51.5341645	0.0	0
1	1	7.4377721	51.4550389	0.0	2
2	1	7.4461245	51.4532095	0.0	100
3	1	7.4565613	51.4680589	0.0	4
...	...	...	...	...	...

Tabelle 7: Beispiel der zweiten Output-Datei des *gen\_csv\_files.py* Programms

Tabelle der Simulationsläufe

	<i>problemName</i>	<i>vehicleID</i>	<i>cutoff-Time</i>	<i>problem-Type</i>	<i>numUAVs</i>	<i>num-Trucks</i>	<i>requireTruckAtDepot</i>	<i>requireDriver</i>	<i>E-type</i>	<i>ITER</i>
python main.py	20212505T131310801008	102	600	1	3	-1	0	0	1	-1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	-	0	-	-	-	-	-
-	-	-	-	2	3	-	-	-	-	1
python main.py	20212505T131310801020	102	600	2	3	-1	0	0	1	1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	1	0	-	-	-	-	-1
python main.py	20212505T131310801050	102	600	2	3	-1	0	0	1	1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	1	0	-	-	-	-	1
python main.py	20212505T131310804008	102	600	1	3	-1	0	0	1	-1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	-	0	-	-	-	-	-
-	-	-	-	2	3	-	-	-	-	1
python main.py	20212505T131310804020	102	600	2	3	-1	0	0	1	1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	1	0	-	-	-	-	-1
python main.py	20212505T131310804050	102	600	2	3	-1	0	0	1	1
-	-	-	-	-	1	-	-	-	-	-
-	-	-	-	1	0	-	-	-	-	-1

Tabelle 8: Simulationsläufe und Argumente