Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

Frida Johanne Kolstad Erdal
Katrine Pedersen
Mathilde Marie Solberg

Navigating Urban Skies: A Novel Path Planning Approach for Systems of Drones

Reducing Social Impact through a Hybrid Evolutionary Algorithm

Master's thesis in Industrial Economics and Technology Management
Supervisor: Henrik Andersson
Co-supervisor: Tobias Andersson Granberg
June 2023

NTNU
Norwegian University of
Science and Technology

Frida Johanne Kolstad Erdal
Katrine Pedersen
Mathilde Marie Solberg

Navigating Urban Skies: A Novel Path Planning Approach for Systems of Drones

Reducing Social Impact through a Hybrid Evolutionary Algorithm

Master’s thesis in Industrial Economics and Technology Management
Supervisor: Henrik Andersson
Co-supervisor: Tobias Andersson Granberg
June 2023

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Preface

This thesis concludes our Master of Science in Industrial Economics and Technology Management at the Norwegian University of Science and Technology (NTNU). It is the result of the course TIØ4905 - Managerial Economics and Operations Research, Master's Thesis, and was written in the spring of 2023. This thesis builds on our preparatory research project, which we conducted in the fall of 2022.

We are sincerely grateful for the invaluable guidance and feedback we have received from our supervisors Professor Henrik Andersson from the Norwegian University of Science and Technology, and Professor Tobias Andersson Granberg from Linköping University. Furthermore, we highly appreciate the cooperation and discussions we have had with representatives from the AiRMOUR project and LuxMobility. We would also like to thank them for allowing us to assist in the data collection for a study in Stavanger and for the data from various studies that they have shared with us.

June 11, 2023
Trondheim

Frida J. K. Erdal Katrine Pedersen Mathilde M. Solberg

Frida Johanne Kolstad Erdal Katrine Pedersen Mathilde Marie Solberg

Abstract

Unmanned aerial vehicles (UAVs) have the potential to transform society's perception of accessibility. During the last decade, UAVs, also known as *drones*, have accumulated considerable interest from both institutional and commercial actors. UAVs are able to exploit the underutilized low-altitude urban airspace and provide increased flexibility in parcel and passenger transport compared to more traditional means of transportation. UAVs can hence broaden the general population's access to various essential and non-essential goods and services.

It is not unlikely that UAVs will handle a significant share of parcel and passenger transport within a wide range of sectors in the coming years. To make this transition successful, the flight paths taken by each UAV must be well planned and the social impact of the paths must be accounted for. Thus, this thesis proposes a path planning model to generate socially acceptable system flight plans for a set of UAVs in an urban environment.

This thesis studies the multi-UAV path planning (MUAVPP) problem in a static, three dimensional, urban environment. The problem is modeled as a multi-objective mixed integer nonlinear program that balances four objectives which are all to be minimized; flight time, risk, visual pollution and noise pollution. A hybrid evolutionary algorithm (HEA) that combines the non-dominated sorting genetic algorithm III (NSGA-III) with ant colony optimization (ACO) is proposed to solve the MUAVPP problem.

Several gaps in existing literature are addressed in this thesis. Most prominent is the current lack of incorporating visual and noise pollution, which are both critical for social acceptability. Furthermore, this thesis models air risk, in addition to ground risk, which is also absent in the reviewed literature. In addition, the proposed model allows various UAV and operational characteristics to vary between different parts of the UAVs' voyages. Taking a holistic view, the use of an evolutionary algorithm to solve a multi-objective, multi-UAV path planning problem in a three dimensional urban environment, is a novel proposition.

By applying the HEA to the static MUAVPP problem, we see promising results. Using the Norwegian city of Stavanger as a case study, the HEA finds a set of system flight plans that successfully trades off the four objectives for the designed test instances. The HEA works with a large set of system flight plans during its runtime, and selects five diverse system flight plans to return. The solution method thus ensures a manageable number of options for a decision maker to choose from, while simultaneously allowing the degree of social impact to be controlled. The results thus make it clear that it is possible to incorporate social acceptability when performing UAV path planning in urban environments. The proposed model suffers from extensive runtimes however, and future research should focus on efficiency enhancements if the model is to be used in the real world.

Sammendrag

Ubemannede luftfartøy ("unmanned aerial vehicles", UAV) har mulighet til å endre samfunnets syn på tilgjengelighet. I løpet av det siste tiåret har UAVer, også kjent som *droner*, fått større og større interesse fra både institusjonelle og kommersielle aktører. Den lavere delen av luftrommet i urbane områder er underutnyttet, og UAVer kan benytte nettopp denne delen av luftrommet. Samtidig sørger UAVer for større fleksibilitet i transport av pakker og passasjerer sammenlignet med mer tradisjonelle transportmidler. UAVer kan dermed utvide tilgjengeligheten til ulike essensielle og ikke-essensielle varer og tjenester for folk flest.

Det er ikke usannsynlig at UAVer vil håndtere en betydelig del av pakke- og passasjertransporten innen en rekke ulike sektorer de kommende årene. For at denne overgangen skal bli vellykket må flyruten til hver UAV være godt planlagt, og påvirkningen flyrutene har på samfunnet må hensyntas. Formålet med denne masteroppgaven er derfor å foreslå en ruteplanleggingsmodell som genererer sosialt akseptable systemflygeplaner for et sett med UAVer i et urbant miljø.

Denne masteroppgaven studerer fler-UAV ruteplanleggingsproblemet ("multi-UAV path planning", MUAVPP) i et statisk, tredimensjonalt, urbant miljø. Problemet er modellert som et ikkelineært, fler-objektiv, blandet heltallsproblem som balanserer fire objektiver; flygetid, risiko, visuell forurensing og støyforurensing. Alle objektivene skal minimeres. En hybrid evolusjonær algoritme som kombinerer *non-dominated sorting algorithm III* (NSGA-III) med *ant colony optimization* (ACO) brukes for å løse problemet.

Flere hull i eksisterende litteratur adresseres i denne masteroppgaven. Mest fremtredende er den manglende inkluderingen av visuell forurensing og støyforurensing, som begge er svært sentrale faktorer for sosial aksept. I tillegg modellerer denne masteroppgaven både luft- og bakkerisikoen tilknyttet UAVer, hvilket ikke ble funnet i den undersøkte litteraturen. Den foreslåtte modellen tillater også variasjon av ulike karakteristikker ved UAVer og deres operasjoner mellom ulike deler av UAVenes reiser. Fra et mer overordnet perspektiv, er bruken av en evolusjonær algoritme til å løse et fler-objektiv, fler-UAV ruteplanleggingsproblem i et tredimensjonalt, urbant miljø en ny tilnærming.

Ved å anvende den hybride evolusjonære algoritmen på MUAVPP problemet, ser vi lovende resultater. Algoritmen finner et utvalg systemflygeplaner som gjør gode avveininger mellom de fire objektivene for et sett med testinstanser som benytter Stavanger som en casestudie. Algoritmen jobber med en stor mengde systemflygeplaner underveis i kjøringen, og velger ut fem differensierte systemflygeplaner som returneres. Løsningsmetoden sørger dermed for at beslutningstakeren får et håndterbart antall systemflygeplaner å velge mellom, og sikrer samtidig at graden av samfunnspåvirkning kan kontrolleres. Resultatene gir altså en klar indikasjon på at det er mulig å hensynta sosial aksept i ruteplanleggingen for UAVer i urbane miljøer. Den foreslåtte modellen har imidlertid lang kjøretid, og fremtidig forskning bør derfor fokusere på effektivitetsfremmende tiltak hvis modellen skal brukes i den virkelige verden.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACO Ant Colony Optimization.

GA Genetic Algorithm.

HEA Hybrid Evolutionary Algorithm.

MDA Multi-Objective Dijkstra’s Algorithm.

MINLP Mixed Integer Nonlinear Program.

MUAVPP Multi-UAV Path Planning.

NSGA-III Non-Dominated Sorting Genetic Algorithm III.

SPL Sound Pressure Level.

UAV Unmanned Aerial Vehicle.

Chapter 1

Introduction

The low-altitude airspace in urban environments represents a major opportunity for transportation that has yet to be fully utilized. An unmanned aerial vehicle (UAV), commonly called a *drone*, can operate in this airspace. A UAV is an unmanned aircraft that is either autonomous and guided by onboard software or controlled remotely (Hu & Lanzon, 2018; Merriam-Webster, n.d.). Using this definition, the first UAVs were developed in the 20th century with a military use case, and since then the use cases and interest in UAVs have expanded rapidly (Lutkevich, 2021; Posea, 2021). The European Commission envisions UAVs to provide numerous services by 2030, including emergency and commercial transport services (European Commission, 2022). Delivery UAV trials have taken place in various countries and the first passenger UAV trials are expected to take place in 2023 (European Commission, 2022; Stavanger kommune, n.d.).

The market for UAV services is expected to grow significantly in the coming years (European Commission, 2022; Grandview Research, 2020), implying both possibilities and challenges. The commercial potential of UAVs includes taxi services and parcel delivery from e.g. DHL or Amazon. The medical sector can also benefit greatly from UAV services, through transport of medical personnel or delivery of e.g. organs, blood products or vaccines. More broadly, UAVs can take advantage of the aforementioned opportunity in the low-altitude airspace, and thus become a middle ground between transport in high-altitude airspace and transport on the ground. UAVs avoid the limitations road infrastructure imposes on vehicles on the ground. Furthermore, UAVs do not need as much space for take-off and landing as airplanes, which again gives the UAVs greater maneuverability and access to various areas. Despite the possibilities, current regulations heavily constrain the use of UAVs in urban environments, making it rather limited. Widespread use of UAVs also introduces challenges. Studies by the AiRMOUR-project found that people have concerns regarding the risk, visual pollution and noise pollution associated with more widespread use of UAVs. AiRMOUR is a research and innovation project focused on the use of UAVs for emergency medical services in urban areas, and is supported by the European Union (EU)(AiRMOUR, 2022).

Path planning is key to effective UAV transport. Path planning may be defined as finding optimal paths between locations (Gasparetto et al., 2015) and many variations of the problem have been studied. It has been used to gather information, to reach a given destination or to visit a set of locations. Path planning has also been studied for various vehicles, both in single-vehicle and multi-vehicle settings, including UAVs, ships, robots and trucks. Minimizing time or distance traveled is a commonly studied objective, and both single-objective and multi-objective models have been proposed. Among the multi-objective models, a bi-objective model of risk and either time or distance is often used. The environments studied also vary greatly, with various combinations of the four dichotomies; two or three dimensional, static or dynamic, discrete or continuous and urban or rural. Evolutionary algorithms (EA) are often proposed to solve path planning problems, but many also develop their own problem-specific exact methods or heuristics, often based on shortest path algorithms.

Motivated by the great opportunity of navigating urban skies, the purpose of this thesis is to develop a path planning model to generate *system flight plans* for a set of UAVs that accounts for the challenges associated with UAV transport. We define a *system flight plan* as a set with one flight path with associated timestamps for each UAV in the system. Each UAV has one or more locations that it must visit in sequence, in addition to its start location. The environment studied is three dimensional, static and urban, and it is discretized and represented by a grid. The problem may thus be defined as a static, multi-UAV path planning (MUAVPP) problem. The problem considers four objectives; flight time, risk, visual pollution and noise pollution. This thesis considers the risk UAVs pose to humans if they were to fall down due to malfunctions or collisions, as opposed to the more common consideration of the risk of the UAVs being destroyed. Since all objectives are to be minimized, the MUAVPP problem is a multi-objective minimization problem. The problem also limits the energy consumption of each UAV to avoid exceeding its energy capacity. This thesis solves the problem using a hybrid evolutionary algorithm (HEA) consisting of non-dominated sorting genetic algorithm III (NSGA-III) and ant colony optimization (ACO). The HEA is run on various test instances to prove or disprove a set of hypotheses for the MUAVPP problem in the computational study.

This thesis contributes to current literature in multiple ways. To the best of our knowledge, visual pollution and noise pollution have not been taken into account in any path planning problem before. Furthermore, the risk to humans associated with UAVs flying in urban environments has also received little attention in the literature. These aspects are crucial to urban

UAV path planning, as widespread use of UAVs will affect the population in urban areas. Thus, by taking risk, visual pollution and noise pollution into account, this thesis aims to generate socially acceptable system flight plans. Another contribution is the possibility to let characteristics differ between a UAV's *segments*. A segment consists of a pair of subsequent locations on a UAV's voyage. Hence, if a UAV is to visit locations A, B and C, there are two segments, namely A to B and B to C. This thesis allows the importance of efficiency, i.e. the importance of short flight time, and the weight of the payloads carried by a UAV to vary between segments. Furthermore, this thesis contributes through the proposed solution method, since an evolutionary algorithm has, to the best of our knowledge, not been used to solve a multi-objective, multi-vehicle problem in an urban environment before.

The remainder of this thesis is structured into nine chapters. Chapter 2 introduces relevant background information regarding UAVs, applicable laws and regulations, urban environments, risk, visual pollution and noise pollution. Chapter 3 presents literature relevant to the MUAVPP problem. Chapter 4 gives a detailed description of the MUAVPP problem, which is used to formulate the mathematical model of the problem in Chapter 5. Chapter 6 presents the models for the environment, energy consumption, risk, visual pollution and noise pollution associated with both a single UAV and a system of UAVs, while Chapter 7 proposes a solution method to solve the MUAVPP problem. Chapter 8 is the computational study, and it presents the input parameters, hypotheses and test instances, and discusses the results of applying the solution method to the test instances. Chapter 9 provides concluding remarks, and Chapter 10 discusses possibilities for future research.

Chapter 2

Background

This chapter presents relevant background information. Section 2.1 gives a simplified overview of UAV characteristics, while Section 2.2 introduces different UAV use cases. Section 2.3 gives an overview of laws and regulations to give an understanding of the limitations currently imposed on UAV usage, while Section 2.4 introduces our definition of an urban environment. Sections 2.5-2.7 describe factors that are taken into account when evaluating different UAV paths in this thesis. Section 2.5 explains the risks associated with UAVs, while Sections 2.6 and 2.7 describe the concepts of visual and noise pollution, respectively. This chapter is a revised and updated version of the corresponding chapter in our preparatory research project.

2.1 Unmanned Aerial Vehicle Characteristics

The basic functionality of a UAV is flight and navigation. In order to fly, UAVs have a frame, rotors and a power source. The frame tends to be a composite and lightweight material to ensure maneuverability. The two main categories of UAVs are rotor and fixed-wing. The rotor category includes both single-rotor and multi-rotor, such as quadcopters. UAVs may also include many other components, such as speed and flight controllers, GPS, various sensors, antennas, cameras, receivers, an altimeter to measure altitude and an accelerometer to measure speed (Lutkevich, 2021). Due to numerous possible components, UAVs may vary significantly in design, e.g. in size, weight, speed, emitted sound, battery capacity and flight range.

2.1.1 Energy Consumption

Both external and internal factors affect energy consumption and thus the flight range of a UAV. The factors can be categorized into three groups; *UAV design*, including weight and size, *UAV dynamics*, such as speed, flight altitude and type of motion, and the *environment*, such as weather and air density (Demir et al., 2014; Zhang, Campbell et al., 2021). Empirical studies of UAV energy consumption have looked at many of the factors in these three categories. For example, Abeywickrama et al. (2018) found that energy consumption increases with increasing speed and with increasing height. When a UAV is hovering at a fixed altitude, a higher altitude leads to a higher energy consumption, and flying horizontally at a given altitude is more energy consuming than hovering at the same altitude. Furthermore, flying vertically upwards requires more energy than flying vertically downwards. Increases in the weight and potential payload weight of a UAV were also found to increase energy consumption.

2.2 Unmanned Aerial Vehicle Use Cases

The first UAVs were used by the military. In later years the use of UAVs has expanded and today UAVs also take on many different civilian use cases. Some examples include delivery, search and rescue, monitoring of traffic, wildlife and weather, surveillance, medical and emergency services, passenger transport, photography and hobbyist use (Lutkevich, 2021; Posea, 2021). The remainder of this section takes a closer look at commercial and medical use cases.

2.2.1 Commercial

Using UAVs for delivery of commercial parcels has been in the works for a while. Since 2013, companies like Amazon, UPS, Alphabet and DHL have all been developing their own UAV technology for parcel delivery and urban logistics (Roca-Riu & Menendez, 2019). There are many different kinds of parcels that may be delivered, including food, clothing, equipment and other goods. Alphabet's Wing initiative has tested UAV delivery of small parcels, such as food, beverages and over-the-counter chemist items, in multiple regions. DHL's Parcelcopter has been used to deliver parcels to rural areas in Germany (Burgess, 2016). In addition, UAVs have been tested in combination with trucks to deliver e-commerce parcels, with the trucks working as mobile landing platforms for the UAVs (Roca-Riu & Menendez, 2019).

The use of UAVs for passenger transport is a novel concept. Multiple actors are currently competing to be the first to offer a UAV passenger service, with the largest barriers being related to regulations and safety (Feist, 2022; Sims, 2022). In 2021, Jetson became the first European company, and among the first companies in the world, to produce a passenger UAV for personal use (Cluff, 2021; Konopka, 2022). EHang is another company making passenger UAVs, including a taxi UAV for inter-city travel with two seats (Singh, 2021b). The US-based company Archer plans to launch a two-seat taxi UAV by 2024 (Singh, 2021a). An example of a passenger UAV is shown in Figure 2.1.

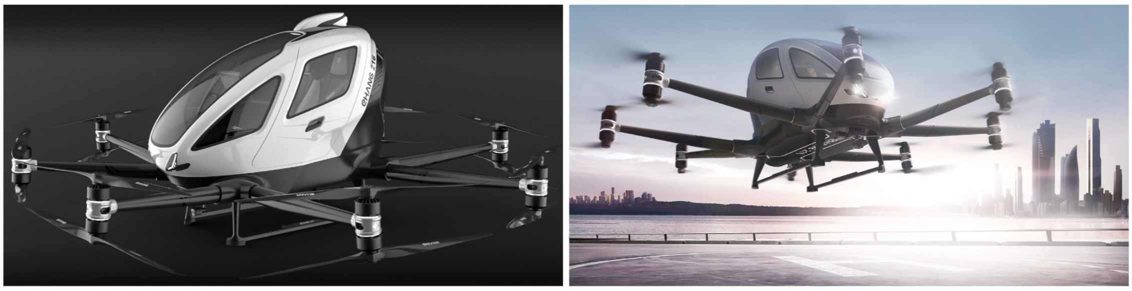


Figure 2.1: The EHang 216 Autonomous Aerial Vehicle (EHang, 2016).

2.2.2 Medical

The first use of medical UAVs were for damage assessments in disaster areas, as UAVs could avoid roadblocks and fly over difficult terrains without endangering the life of a flight crew (Meier, 2015). They have also been used to assess chemical or biological hazards, measure radiation, aerosols and other health hazards, track disease spread and gather information about the number of people in high-risk environments that are in need of care (Rosser et al., 2018).

In addition to assessments, a large part of UAVs' role in medical services relates to delivery. UAVs was used for delivery of aid packages to people affected by disasters as early as in 2010, and have delivered disease testing kits to, and transported test samples from, remote villages and rural areas in Africa (Balasingam, 2017). Multiple studies have also shown that UAVs can safely transport blood products and vaccines (Balasingam, 2017; Roca-Riu & Menendez, 2019; Toor, 2016). Delivery UAVs have also been tested in the US for transport of medical supplies to a small clinic in a rural part of Virginia (Balasingam, 2017; Rosser et al., 2018).

UAVs can also be used for emergency transport. This includes the delivery of automated external defibrillators (AED) to people suffering cardiac arrests outside of hospitals (Rosser et al., 2018; Schierbeck et al., 2021). This UAV use case has been tested in relation to the AiRMOUR project in Sweden in 2021 and Norway in 2022 (Crumley, 2022; Crumley, 2021), and is already practiced by the company Everdrone in Sweden (Everdrone, n.d.). Passenger UAVs are also used in emergencies to transport medical personnel to people in need (Crumley, 2022). Through the AiRMOUR project, EHang is currently planning to test passenger UAVs in Stavanger in Norway in 2023 (Stavanger kommune, n.d.).

2.3 Laws and Regulations

Due to the novelty of the UAV sector, there is still ongoing work on UAV regulations. In 2019, the European Commission adopted several UAV regulations, focusing mainly on individual UAV operations (European Commission, 2019a; European Commission, 2019b). Due to the increasing complexity and number of UAV operations entering the airspace, the European Commission adopted another set of regulations in 2021, with additional rules and procedures regarding certain geographical zones, called the *U-space* (European Commission, 2021a; European Commission, 2021b; European Commission, 2021c). These zones for example include areas where a large number of simultaneous UAV operations are expected. The regulations adopted by the European Commission are the foundation for all UAV operation regulations in the EU and a number of other countries, including Norway.

In addition to the regulations described above, countries can have individual, supplementary regulations. The most important regulation for UAVs in Norway is the "regulation on unmanned aircraft" ("Forskrift om luftfartøy som ikke har fører om bord"), also known as "The drone regulation" ("Droneforskriften") (Samferdelsdepartementet, 2015). In addition, UAV traffic is regulated through the law of aviation ("Luftfartsloven") (Samferdelsdepartementet, 1996) as well as through a number of other laws and regulations.

There is considerable overlap between the contents of the various UAV regulations. All the regulations establish three different categories of UAV operations, affecting what kind of rules that apply. The regulations mostly include rules and procedures regarding the competency of a remote pilot, the airworthiness of the UAV and the authorization and registration of the UAV operation. In addition, there are rules regarding the allowed flight altitudes, as well as procedures for risk management and noise minimization. The regulations also specify that there are some areas that can be particularly sensitive to UAV operations and that local authorities may enforce additional rules that prohibit or restrict UAV operations in certain areas. These types of areas are referred to as *no-flight zones* in this thesis. Examples of no-flight zones are airports, embassies and prisons. Furthermore, the EU regulations on U-space contain requirements that ensure collaboration between the different stakeholders of a U-space, e.g the U-space service providers, the UAV operators and relevant air traffic service units (European Commission, 2021a; European Commission, 2021b; European Commission, 2021c). Some examples of requirements are specific procedures for coordination and communication, such as geo-awareness services and weather information services. With increased use of the U-space, a major objective is to avoid collisions between aircrafts. The Standardized European Rules of the Air (SERA) is based on the pilot of an aircraft using the "see and avoid" principle, to avoid mid-air collisions. The European Commission points out that since UAVs are by definition unmanned, this principle cannot be strictly followed. Hence there is a need for new ways to mitigate the risk of collisions between UAVs (European Commission, 2022).

The rapid development of UAV technology and expansion of UAV use cases makes it difficult to know what future UAV regulations will look like. As society learns more about the effects of widespread use of UAV operations, a need for new and improved regulations arise. As of May 2023, UAVs that operate over assemblies of people or carrying passengers, are so heavily regulated that such UAV operations are in practice prohibited. In addition, the regulations are heavily based on the fact that there is a pilot controlling the UAV remotely, with little attention given to UAVs guided by onboard software. Therefore, a continuous development of UAV laws and regulations is a big part of the UAV strategy of the European Commission from November 2022 (European Commission, 2022).

2.4 The Urban Environment

There exist several definitions of an urban environment, and the European Commission points out that it can be hard to give a definition that applies globally (Dijkstra & Poelman, 2014). Nevertheless, most uses of the term *urban environment* emphasize a high population density (Rutledge et al., 2022). Cities and their surrounding areas are thus natural examples of urban environments. The term urban environment is important for research and regulations of UAVs, because a high population density implies a higher risk associated with UAV operations in these areas. Current regulations strongly limit the use of UAVs in urban areas (European Commission, 2019a; European Commission, 2019b), but with smart city projects, such as the Smartbyen project in Stavanger (Stavanger kommune, n.d.), urban UAV usage is expected to become increasingly common in the future.

2.5 Risk

It is widely acknowledged that risk assessments of UAV operations are essential (EASA, 2015; Primatesta et al., 2020; Sedov, Polishchuk & Vishwanath, 2021; Watkins et al., 2020). This is also supported by the fact that the regulations adopted by the European Commission in 2019 follow a risk-based approach (European Commission, 2019a; European Commission, 2019b). The importance of risk is also emphasized in the official UAV strategy of both Norway and the EU (European Commission, 2022; Helse- og omsorgsdepartementet et al., 2018). A central goal of UAV operations is the minimization of air risk and ground risk (European Commission, 2019a; European Commission, 2019b; Helse- og omsorgsdepartementet et al., 2018). *Ground risk* comprises third-party risks, i.e. the risk to people, animals and property on the ground which are not a part of the UAV operation. *Air risk* is the risk for users of the same airspace as a UAV. Air risk comprises the risk for the passengers inside the aerial vehicles, both in the UAV itself and in other vehicles, due to the possibility of the UAV colliding with other aerial vehicles or malfunctioning (Helse- og omsorgsdepartementet et al., 2018; Samferdelsdepartementet, 2015; Sedov, Polishchuk, Maury et al., 2021; Sedov, Polishchuk & Vishwanath, 2021).

As of May 2023, the UAV risk assessments in Europe are mainly qualitative. UAV operations of a certain risk level must conduct an operational risk assessment to get an approval to carry out the operation (European Commission, 2019a; European Commission, 2019b; Sedov, Polishchuk & Vishwanath, 2021). A common way to quantify and measure the risk associated with UAV operations, is a probabilistic approach calculating the *expected fatality rate* (EFR) of a given path (Primatesta et al., 2020; Sedov, Polishchuk, Maury et al., 2021; Sedov, Polishchuk & Vishwanath, 2021; Susini, 2015).

2.5.1 Air Risk

Air risk is the primary focus of risk assessments for conventional aviation traffic, and conventional aviation traffic has therefore inspired much of the work on air risk associated with UAV operations (Primatesta et al., 2020; Sedov, Polishchuk, Maury et al., 2021; Sedov, Polishchuk & Vishwanath, 2021). However, unlike conventional aviation, UAVs do not have a clearly defined risk measurement (EASA, 2013). Sedov, Polishchuk, Maury et al. (2021) calculate the air risk of a UAV path as the EFR associated with the path, using the expected number of conflicting airplanes along the path and the expected number of people on board each airplane. The work shows that the EFR associated with air risk is virtually zero in most places, because UAVs and airplanes operate at different altitudes. The only exception is near airports, but since airports are considered no-flight zones for UAVs, as described in Section 2.3, the increased EFR in these areas is irrelevant. Sedov, Polishchuk, Maury et al. (2021) and other work in the field (Helse- og omsorgsdepartementet et al., 2018; Primatesta et al., 2020; Sedov, Polishchuk & Vishwanath, 2021), do not take passenger UAVs into account due to their novelty. Therefore, the air risk for the people on board a UAV is not considered. With increased use of the U-space described in Section 2.3, the risk of collisions between UAVs must be taken into account.

2.5.2 Ground Risk

Ground risk is a fairly new research area, motivated by the fact that UAV operations often operate in closer proximity to people compared to e.g. conventional aircraft (Sedov, Polishchuk, Maury et al., 2021). Various quantifications of risk and calculations of EFR have been proposed. The EFR calculations often depend on the probability that a UAV falls down, the probability that the UAV hits people on the ground and the probability of fatality if the UAV hits a person. These factors are again dependent on the characteristics of the UAV, such as its weight, size and battery capacity, but also external factors such as the population density, the weather and the sheltering level of the overflown area (Primatesta et al., 2020; Sedov, Polishchuk & Vishwanath, 2021; Susini, 2015).

2.6 Visual Pollution

There have been many different definitions of visual pollution over the years, with the definitions gradually becoming wider (Jensen et al., 2014; Portella, 2007; Sumartono, 2009; Yilmaz & Sagsöz, 2011). A general definition of visual pollution is the negative impact an individual may experience by viewing a visual pollutant and its movement. A visual pollutant may be defined as any object or artificial structure that degrades visual quality. This includes any element that may be considered as misplaced in a given outdoor or indoor landscape, and also relates to the spatial arrangement of elements and whether they appear to be organized (Ahmed et al., 2019; Thomas, 2022). Examples of visual pollutants thus include littered waste, advertisements, buildings that stand out from surrounding infrastructure, unordered parking spots, wires and wind farms (Mohamed et al., 2021; Skenteris et al., 2019). A UAV flying through a city could thus also be considered as a visual pollutant, and widespread UAV usage could further amplify the pollution (Kraus et al., 2020). There exists various methods for *studying* visual pollution (Thomas, 2022). Despite the different methods, visual pollution is difficult to *quantify* as little research exists and visual pollution is subjective (EASA, 2021).

Visual pollution can have various health-related consequences, such as distraction, mood disorders, stress, anxiety, overstimulation and reduced work efficiency (Mohamed et al., 2021; Thomas, 2022). Furthermore, visual pollution does not fade over time, and can even increase the perception of other pollutants such as noise, since viewing the source of noise pollution may amplify a person's perception (Schäffer, Pieren, Hayek et al., 2019; Thomas, 2022). Despite these possible consequences of visual pollution, it was rated second to last out of ten proposed reservations towards delivery UAVs in a European Union Aviation Safety Agency (EASA) survey, behind e.g. noise pollution and risk (EASA, 2021). It should be noted however, that visual pollution was not considered to be insignificant despite being rated as less important. Thomas (2022) found that the four factors that influence visual pollution the most are distance to the closest UAV, the number of UAVs, the UAVs' use cases and the awareness of the UAVs. Whether the environment was urban or rural did not have much of an impact.

Visual pollution can be reduced through various techniques. Generally, UAVs should avoid flying in open and highly populated areas. If it cannot be avoided, the UAVs should fly at high altitudes to appear smaller in size. Another possible measure is to highlight the positive aspects of using UAVs, since having a positive opinion prior to seeing UAVs lead to people perceiving UAVs as less visually polluting (Thomas, 2022). This also coincides with our findings in Appendix A. Finally, since the studies in Appendix A indicate a greater acceptance for emergency medical UAVs, they should be clearly marked, e.g. with an identifying color.

2.7 Noise Pollution

According to Jhanwar (2016), noise pollution is distressing sound that may harm the physical or mental activity of humans or animals. In this thesis, the word *sound* is used to describe any audio, while the word *noise* is considered as a subcategory of sound, describing an undesirable and negatively associated sound. Using the classes of noise pollution proposed by Jhanwar

(2016), UAV noise may be classified as environmental noise, and is characterized by high frequencies and high tonality (Alexander & Whelchel, 2019; Torija, Self et al., 2019). A UAV emits noise through its engine but also aerodynamic noise from compression and friction in the air when the UAV is in motion (Jhanwar, 2016).

Psychological factors impact the perceived noise pollution, and people may have different subjective perceptions of the same noise (Torija & Clark, 2021). Moreover, a 2017 study from NASA shows that UAV noise is considered more annoying per decibel (dB) compared to noise from other forms of transportation. Possible explanations for this are that noise coming from above can be perceived as more frightening, that the noise emitted by UAVs is unfamiliar or that UAV noise might remind people of insects (Christian & Cabell, 2017; Norsk forening mot støy, 2018). Furthermore, an individual can perceive the same UAV noise differently depending on the presence of other background noise, since such a presence affects the noticeability of the UAV noise. It might thus be beneficial to make UAVs fly above highways and other areas where there is already a high noise level (European Commission, 2022).

There exist many different metrics and measurement methods for noise, emphasizing different sound characteristics. However, it is not generally agreed upon which procedures are best for UAV noise, among other things because UAVs have characteristics that make their noise different from the noise emitted by more researched vehicles, such as airplanes (Schäffer, Pieren, Heutschi et al., 2021; Torija & Clark, 2021). EASA has worked on creating guidelines for objective and comparable noise measurements of UAVs for years, and the agency published a first draft in October 2022 (EASA, 2022). It is likely that there will be more research on the topic as the use of UAVs become more widespread.

Noise pollution is a major concern in the discussion of widespread use of UAVs, and it plays a role in the development of UAV regulations, as mentioned in Section 2.3. In general, the lower the sound level, the better and the overall aim is hence to minimize noise. The World Health Organization (WHO) has published guidelines of acceptable noise levels from various noise sources, and different countries and areas can have specific limits on acceptable levels of noise (WHO, 2018). Noise pollution can have adverse health effects, and is thus important to consider when deciding where UAVs may or may not fly. According to Aasvang et al. (2022) at Folkehelseinstituttet (FHI), noise above 80-85 dB over a long time period can result in permanent damage. It is also well documented that noise can disturb concentration and sleep. The long-term effects of noise and its relation to diseases is more uncertain, but it seems to increase the likelihood for some heart and vascular diseases (Aasvang et al., 2022). There is, however, ongoing research on possible methods to reduce UAV noise, and for example is the company Whisper Aero working on designing a UAV that customizes its sound so that it blends into the sound in its surroundings (Vance, 2021).

Chapter 3

Literature Review

This chapter presents a review of relevant literature for the static, multi-UAV path planning (MUAVPP) problem studied in this thesis. Path planning is a fundamental part of UAV operations and has been researched for decades (Bortoff, 2000). When also including other vehicles, path planning has been studied even longer and the amount of literature is vast. This chapter presents the key findings and is split into seven sections. Section 3.1 describes the search strategy used. From the large set of literature initially gathered using the search strategy, 36 papers were selected for further analysis. Sections 3.2-3.6 classify these papers based on various aspects. In Section 3.2 the problem categories and problem types are presented, while Section 3.3 discusses the model elements found in the papers. Sections 3.4 and 3.5 introduce the environments and vehicles studied, while 3.6 discusses the solution methods used. Finally, Section 3.7 takes a holistic view of the classifications from the previous sections, discusses identified gaps in relation to the various aspects and explains our contribution to the existing literature. The majority of the literature review in this thesis was conducted in relation to our preparatory research project, but has been updated to fit the MUAVPP problem.

3.1 Search Strategy

The search for literature started out broad, and was later narrowed down based on relevance. Since there exists numerous variants of path planning and related problems, with differing considerations and approaches, it was deemed necessary to use a wide set of search terms. The search terms are presented in Table 3.1.

To ensure a structured literature review, the search terms were combined into search queries using logical operators such as "AND" or "OR". A query could contain multiple search terms from the same group, e.g. “Multi UAV”, and a query did not necessarily contain a term from

Table 3.1: Overview of search terms used in various combinations to find relevant literature.

Problem	Model elements	Environment	Vehicle	Solution method
Path planning	Energy consumption	Continuous	Drone	Exact
Route planning	Multi-objective	Discrete	UAV	Heuristic
Shortest path	Noise pollution	Dynamic	Robot	Mathematical programming
Trajectory planning	Risk	Static	Ship	Optimization
	Visual pollution	Urban	Truck	
		2D	Single	
		3D	Multi	

every group. The queries were then fed into *Scopus*, a search engine for a database of citations and abstracts. We have also utilized *ScienceDirect* and *Google Scholar* to further broaden our search. During the search and review process, the pool of literature was further extended to include papers citing, and papers cited by, the papers found.

This search strategy resulted in a large number of papers that were carefully assessed. By sorting out the papers most relevant for this thesis, we were left with 36 papers, which are analyzed from multiple angles in the remaining sections. The selected papers are organized in Table 3.2 in Section 3.7.

3.2 Problems

The papers in this literature review studies various *problem categories* and *problem types*. We define the problem category as the overarching problem that is solved. Three recurring problem categories are path planning, trajectory planning and route planning. Both path planning and trajectory planning aim to find one or more optimal paths between a set of ordered locations. Path planning differs from trajectory planning by discretizing the environment and defining an optimal path as a set of points through a grid or other type of network, limiting the possible paths a vehicle may take. Trajectory planning, on the other hand, describes an optimal path by a time schedule of when and how much to update the trajectory the vehicle is currently on. Thus, there is no limit to the possible paths a vehicle may take when using trajectory planning (Gasparetto et al., 2015). The route planning problem differs more clearly from the previous two problem categories, as route planning finds the *optimal order* to visit a set of locations (Murray & Raj, 2020). Hence, path planning and trajectory planning focus on finding the actual path between given, ordered locations, whereas route planning focuses on deciding the visitation order of locations. Of the 36 papers analyzed, about 69% study path planning, 19% study trajectory planning, and 25% study route planning, often combined with either path or trajectory planning.

In addition to an overarching problem category, most of the papers have a more specific purpose or *problem type*. Three recurring problem types are observed in the reviewed papers; information gathering, reaching a target and visiting a set of locations. The information gathering problem type focuses on gathering as much information as possible in a given area, but does not require that one or more locations *must* be visited. The information gathering problem type is used in 19% of the papers, and includes surveillance (Modares et al., 2017), search and rescue (Hayat et al., 2017), identifying causes of disasters (Qadir et al., 2022) and other forms of data gathering. The second problem type is simply to reach a target location from a given initial location, and is found in 33% of the papers. The third problem type is to visit a set of given, ordered or unordered locations, possibly for the purpose of delivery, and is found in 36% of the papers. The problem type of visiting locations may be viewed as a set of consecutive reaching-a-target problems between two and two locations. We refer to a set of two consecutive locations as a *segment*. The three problem types are shown in Figure 3.1.

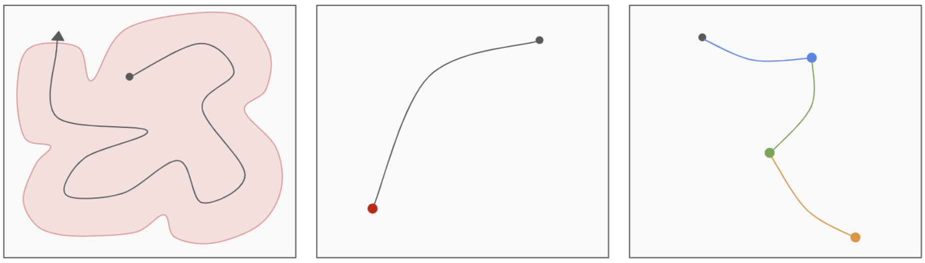


Figure 3.1: Illustration of the three problem types; the left part is information gathering, the middle part is reaching a target and the right part is visiting a set of locations. The gray dot is the starting point for each of the problem types. In the left part of the figure, the red shaded area around the actual path is the area information was gathered from. In the right part of the figure, there are three segments, shown by the blue, green and orange lines.

3.3 Model Elements

The analyzed papers have included various elements as objectives or constraints in their models. Some common elements include time, distance, risk and energy consumption. Almost 69% of the analyzed papers include either time or distance minimization in the objective function, which means they are essentially modeling versions of the well-known shortest path problem.

Risk minimization is an objective in 39% of the papers. Among the papers that study UAVs and have risk as an objective, only Sedov, Polishchuk and Vishwanath (2021) considers the ground risk to humans, as defined in Section 2.5. The rest of the papers focus solely on the risk of the UAVs being destroyed. Destruction can for example happen through collision with static or dynamic obstacles or by being shot down by enemies. Collision avoidance has been included both as an objective, e.g. by Shao et al. (2020), and as a constraint, e.g. by Bai et al. (2022). All but one of the papers that include collision avoidance incorporate it through a measurement of the closest distance between the objects that are at risk of colliding. Furthermore, only Fügenschuh and Müllenstedt (2015) differentiate the collision risk based on the type of vehicle or object that the vehicle in question could collide with.

Energy consumption is included as a minimization objective in 19% of the papers. Papers with energy consumption as an objective tend to develop more complex and realistic models for energy consumption compared to those only including energy consumption as a capacity constraint. The latter often opt for modeling energy consumption as a function of only time or distance traveled, while the prior often also include kinematic characteristics of the vehicle's movement, such as speed.

There is an even split between single-objective and multi-objective problems in the papers. The most common multi-objective model is a bi-objective model minimizing risk and time or distance traveled. Some of the papers also have three or more objectives, including Hu, Naeem et al. (2017) and Ma et al. (2018). All of the papers minimizing time or distance in combination with other objectives are essentially studying multi-objective shortest path problems.

3.3.1 Multi-Objective Optimization

Following Jaimes et al. (2009), a multi-objective optimization problem (MOP) is defined as

$$\begin{aligned} Minimize \quad & \vec{\boldsymbol{f}}(\vec{\boldsymbol{x}}) = [f_1(\vec{\boldsymbol{x}}), f_2(\vec{\boldsymbol{x}}), ..., f_k(\vec{\boldsymbol{x}})] \\ s.t. \quad & \vec{\boldsymbol{x}} \in \mathcal{X} \end{aligned}$$

where $\vec{\boldsymbol{x}}$ is a vector of n decision variables, $\mathcal{X} \subseteq \mathbb{R}^n$ is the feasible set, implicitly determined by a set of constraints, and $\vec{\boldsymbol{f}} : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is a vector function made up of k scalar objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$.

While comparing solutions is trivial for single-objective problems, there is no straightforward way to do such comparisons for multi-objective problems. This is for example due to conflicting objectives or differing units for the objectives. A common method for comparing multi-objective solutions is to use the *Pareto dominance relation*. This method does not usually lead to a single solution, but instead finds a set of solutions, called *Pareto optimal solutions*. These solutions have different trade-offs among the objectives, but are all *non-dominated* (Jaimes et al., 2009). For a minimization problem, the Pareto dominance relation states that a vector $\vec{\boldsymbol{z}}$ of length k Pareto dominates another k-length vector $\vec{\boldsymbol{u}}$, and we write $\vec{\boldsymbol{z}} \prec \vec{\boldsymbol{u}}$, if and only if

$$\forall i \in \{1, ..., k\} : z_i \leq u_i \qquad and \qquad \exists i \in \{1, ..., k\} : z_i < u_i$$

Using this relation, a set of candidate solutions may be reduced to the set of Pareto optimal solutions. A solution $\vec{x} \in \mathcal{X}$ is considered *Pareto optimal* if there does not exist a solution $\vec{\mathbf{x}}' \in \mathcal{X}$ that Pareto dominates $\vec{x}$. As shown in Figure 3.2, the set of Pareto optimal solutions in the variable space, $\mathbb{R}^n$, creates a *Pareto front* in the objective space, $\mathbb{R}^k$ (Jaimes et al., 2009).

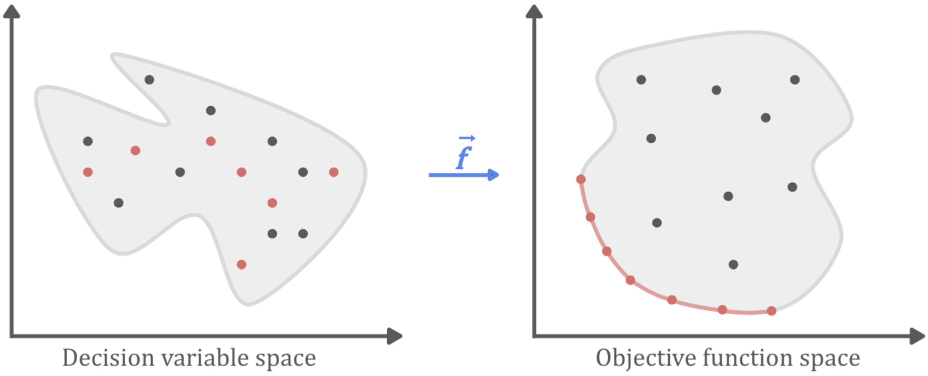


Figure 3.2: Illustration of the Pareto optimal solutions in the variable space and the Pareto front they span out in the objective space. Red dots correspond to Pareto optimal solutions in the left part of the figure, and to objective function values on the Pareto front in the right part. The red lines between the points in the right part of the figure map out the Pareto front.

Despite there being multiple Pareto optimal solutions to an MOP, only one can be selected and implemented in practice. Thus, a decision maker (DM) is needed to provide preference information which may be used to select the most suitable solution. The DM may communicate preferences prior, during or after the search for Pareto optimal solutions. An approach used by half of the MOP papers analyzed in this literature review, including Yang, Tang et al. (2015) and Yang and Yoo (2018), is to minimize the *weighted sum* of all objectives. In this method, the DM conveys their preferences by assigning a weight to each objective and the MOP is converted into a single-objective optimization problem.

3.4 Environments

The environments studied in the papers may be categorized based on different aspects. The first aspect is the representation, which can be either continuous or discrete. In a continuous environment, infinitely many trajectories between two locations are possible. One possible continuous representation of the environment is a *vector of kinematics*, such as speed, angle and acceleration, for each unit of time. Another possibility is to locate *waypoints* where the vehicle changes the trajectory it follows. 22% of the analyzed papers use a vector of kinematics, while 25% use waypoints. For discrete environments, two categories of discretization are found in the papers. The first category is a *grid* representation, which divides the environment into a set of cells using intersecting gridlines (Ren et al., 2019). The grid may either be regular and consist entirely of equidistant parallel or perpendicular gridlines, or it may be irregular.

An irregular grid differs from a regular grid by not containing only equidistant gridlines, but instead allowing gridline sections to be a multiple of other sections. Irregular grids require less memory compared to regular grids, but may result in reduced path quality in areas with large gridline sections (Souissi et al., 2013). About 25% of the papers use regular grids. Only one paper uses an irregular grid, namely Ren et al. (2019). The second category of discretization found in the papers is *node networks*. In such networks, the vertices are not all equidistant nor are all edges parallel or perpendicular to each other. In total, 27% of the papers use a node network environment model.

The second aspect is the dimensionality of the environment. There appears to be more literature considering two dimensional space, with 67% of the selected papers studying movement in two dimensions, while the remaining 33% study a three dimensional space. It is natural to focus on two dimensions when considering vehicles moving on the ground, such as ships or robots. A more surprising finding is that the majority of the papers studying UAVs also use a two dimensional environment, despite the fact that UAVs operate in the air, which is inherently three dimensional. Nonetheless, the percentage of papers using a three dimensional environment *and* studying UAVs is 44%, and hence somewhat higher than the overall percentage of papers studying a three dimensional environment.

The third environment aspect is the static-dynamic dichotomy. In a static environment, all information about the environment is known in advance and no changes occur between decision making and execution. Thus, *offline planning* may be used, meaning all decisions are made before the execution begins (Yang, Tang et al., 2015). Static environments make up 70% of the papers, while the remaining 30% study a dynamic environment. In a dynamic environment, unforeseen incidents may happen during execution. In such environments, *online planning* might be needed to update the decision making during execution. The predominance of static problems may be due to such problems being easier to solve, but it comes at the cost of reduced realism.

The fourth environment aspect is the split between urban and rural environments. Urban environments, as defined in Section 2.4, make up only 21% of the analyzed papers, with all of them studying UAV problems. Rural environments make up 51% of the papers. We have not classified the remaining 28% of the analyzed papers as urban or rural, due to the urban-rural distinction not being as clear for papers studying vehicles other than UAVs.

3.5 Vehicles

Path planning and related problems have been studied for various vehicles. Our search has primarily focused on UAVs, and this type of vehicle thus makes up 72% of the selected papers.

Nonetheless, the problem of path planning has been studied more in-depth in the field of robotics, often under the name of *motion planning*, as found in e.g. Alexopoulos and Griffin (1992) and Li et al. (2011). Other vehicles studied include ships and trucks.

Another important aspect is the number of vehicles. Among the papers, 67% study a single vehicle while the remaining 33% study multiple vehicles. The use of multiple vehicles may indicate that the operations are too complicated for a single vehicle, and often introduces more complexity into the modeling compared to the study of a single vehicle. The multi-vehicle problems studied in the papers either give each vehicle individual tasks, as seen in the paper by Murray and Raj (2020), or make the vehicles work together to complete tasks, as seen in the paper by Hayat et al. (2017). The split between considering heterogeneous and homogeneous vehicles is quite even among the papers studying multiple vehicles.

3.6 Solution Methods

A variety of optimization-based solution methods have been applied to solve the path planning problem and the related problems. This section gives an overview of and discusses the different solution methods found in the papers.

3.6.1 Evolutionary Algorithms

The most common solution method among the analyzed papers is evolutionary algorithms (EA), which make up 44% of the papers. Furthermore, EAs have been used to solve 67% of the papers studying multi-objective problems. EAs are also used in 67% of the papers considering a three dimensional environment, as well as in 44% of the papers considering an urban environment. EAs are stochastic search and optimization methods that in different ways simulate evolutionary processes found in nature. Similar to other stochastic search strategies, EAs are not guaranteed to find optimal solutions, but rather aim to find good solutions in a reasonable amount of time (Jaimes et al., 2009). Two common EAs are genetic algorithms (GA) and ant colony optimization (ACO).

In GAs, a population of candidate solutions is gradually evolved through a set of iterations to increase the *fitness* of the population. In each iteration, pairs of candidate solutions are combined and possibly modified through mutations to create an offspring population. The offspring candidate solutions are then evaluated by a fitness function, which is usually the objective function of the problem. Fit candidate solutions from the offspring population replace less fit candidate solutions in the current population to construct the next generation of the population (Zhao et al., 2018). Some GAs have also been developed specifically for multi-objective problems, e.g. the non-dominated sorting genetic algorithm (NSGA) (Srinivas

& Deb, 1994). NSGA differs from other GAs in the evaluation of candidate solutions, using a non-dominated sorting technique. NSGA-II (Deb, Agrawal et al., 2000), an improvement of the original NSGA, calculates a crowding distance between candidate solutions to ensure diversity when selecting individuals for the next generation. NSGA-II does not scale well with the number of objectives however, which is why NSGA-III (Deb & Jain, 2014) was proposed. Instead of a crowding distance, NSGA-III uses a predefined set of reference points. NSGA-II is used by Ren et al. (2019), NSGA-III is used by Bai et al. (2022), and other GAs are used by e.g. Hayat et al. (2017) and Shivgan and Dong (2020).

The ACO algorithm expresses feasible solutions using the paths of virtual ants. The solution space is represented by the set of ant paths, and over time, the pheromone concentration of good paths will increase. This leads to more ants taking similar paths, and can potentially lead to the whole colony converging to a single best path. This path corresponds to a solution of the optimization problem (Zhao et al., 2018). ACO is used by e.g. Chen, Xu et al. (2017) and Liu et al. (2009), and the algorithm is combined with a GA by Yang and Yoo (2018).

Other EAs found in the papers include particle swarm optimization (Ma et al., 2018; Shao et al., 2020), an algorithm for differential evolution of singular waypoints (Yang, Tang et al., 2015), a Whale optimization algorithm (Wu et al., 2018), a multi-objective EA based on decomposition (Peng et al., 2022) and a dynamic group-based cooperative optimization algorithm (Qadir et al., 2022).

3.6.2 Problem-Specific Heuristics

Problem-specific heuristics are, as the name entails, heuristics. They are thus not guaranteed to find an optimal solution, but will hopefully find good approximations. Problem-specific heuristics make up 22% of the solution methods proposed by the papers, and are commonly used for single-objective problems. Among the studied papers, the paper by Yin et al. (2018) is the only one that uses a problem-specific heuristic for a multi-objective problem. Furthermore, most of the problem-specific heuristics have been developed for two dimensional environments. Among the papers studying a dynamic environment, 36% use problem-specific heuristics. Examples of problem-specific heuristics include an A*-based heuristic in Li et al. (2011), a spatial-temporal clustering-based algorithm in Chen, Du et al. (2021), a heuristic search algorithm in Zhang, Chen et al. (2012) and an adaptation of the Lin-Kernighan heuristic in Modares et al. (2017).

3.6.3 Problem-Specific Exact Methods

In contrast to the problem-specific heuristics described above, problem-specific exact methods find *optimal* solutions to problems and are used in 16% of the papers. Problem-specific exact methods have been used solely on single-vehicle problems in static environments. Most of these papers also study a two dimensional environment and a single objective. Since the static, two dimensional, single-vehicle, single-objective problem is the most simple among those studied, it makes sense that these are the problems that have been attempted to be solved exactly. The problem-specific exact methods used include a variation of A* (Alexopoulos & Griffin, 1992), a graph-based routing algorithm based on the Floyd algorithm (Hentschel et al., 2007), the fast marching square algorithm (Chen, Huang et al., 2020) and a variation of Dijkstra's algorithm (Soltani & Fernando, 2004). Thus, it appears to be common to use shortest path algorithms as the starting point for problem-specific exact methods.

3.6.4 Mathematical Programming

Mathematical programming is a less common solution method and is used in 14% of the papers. It is used in multiple different ways for various problems, environments and vehicles. Chen, Du et al. (2021) and Murray and Raj (2020) use a mixed integer linear program (MILP) to solve small instances of a problem, while a problem-specific heuristic is used for larger instances. Bellingham et al. (2003) use an MILP in a subroutine of a problem-specific heuristic. Only two of the papers solve their problems using only mathematical programming. The first is Chen, Han et al. (2012), who solve a single-UAV trajectory planning problem in a three dimensional, dynamic environment using a linear program for each waypoint. The second is Fügenschuh and Müllenstedt (2015), who solve a multi-UAV routing and trajectory planning problem in a three dimensional, static environment using an MILP.

3.6.5 Other Methods

In addition to the more common solution methods described above, other methods have also been proposed in the papers. Machine learning is used in three of the papers, which all study single-vehicle problems in two dimensional environments (Chen, Huang et al., 2020; Woodley & Acar, 2004; Zaccone & Martelli, 2018). The vehicles in focus in these papers have been either a ship or a truck, and thus no UAVs. Other methods used include a model predictive control algorithm, a logic-based Benders' decomposition, and various dynamic programming methods.

3.7 Literature Overview

From this literature review, it is clear that many variations of path planning and related problems have been studied. At the same time, several gaps can be identified in the literature. Subsection 3.7.1 discusses these gaps, while Subsection 3.7.2 presents how this thesis contributes to filling gaps and creating new insights. Table 3.2 gives an overview of central aspects of the papers analyzed in this literature review. The papers are ordered chronologically by publishing year. This thesis is included at the bottom of the table for comparison.

An interesting observation from Table 3.2, is that studying UAV operations in an urban environment appears to be a quite recent concept, first showing up in papers from 2018. This intuitively makes sense, as urban UAV use cases are relatively new. We also notice that the number of objectives studied increases over time. Another thing to note is that the use of EAs becomes more prominent over time, possibly as the problems increase in complexity. Apart from these developments, there does not appear to be any other clear trends in the literature.

3.7.1 Gaps in the Literature

The literature review reveals various gaps in the current literature. As described in Section 3.2, the most common problem type among the selected papers is to visit a set of locations. However, none of the papers with this problem type look at the possibility of varying characteristics between segments, i.e. between pairs of subsequent locations to be visited. Examples of characteristics that may be varied include vehicle characteristics, such as increasing or decreasing the total weight of the vehicle based on whether the vehicle carries a payload on a given segment, or varying the importance of an objective on different segments. While papers studying heterogeneous vehicles could allow for something similar by having different vehicles travel on different segments, no papers study the possibility of the *same* vehicle changing characteristics between segments.

Despite a variety of objectives and model elements found in the papers, no paper that models visual pollution or noise pollution, either in the objective function or otherwise, was found. While it could make sense to neglect these pollutants in rural environments, such a simplification is less intuitive in an urban environment. Hence, there appears to be a gap in the literature in regards to taking visual and noise pollution into account in path planning and related problems. Furthermore, none of the reviewed papers studying UAVs consider the air risk for passengers of a UAV. This is natural as none of the papers consider passenger UAVs, but is important to include when looking at passenger transport. The consideration of ground risk to humans was also found to be sparse.

Section 3.4 shows that less attention has been given to three dimensional environments compared to two dimensional environments, even among the papers studying UAVs. Furthermore, none of the selected papers study the problem category of path planning for multiple UAVs in an urban environment. Hence, although these aspects are studied in detail separately, there is a gap in the literature as to study the overall effect of this combination.

The reviewed papers use a wide range of solution methods, with EAs being a popular choice. However, none of the papers use an EA to solve a multi-objective, multi-vehicle problem in an urban environment. This thus constitutes a gap.

3.7.2 Our Contribution

This thesis addresses several gaps in the current literature, related to both modeling choices and solution method. An important contribution to the problem type of visiting a set of locations is that this thesis allows for characteristics to vary between segments of a voyage traveled by the same UAV. The weight of the potential UAV payload may for example vary, resulting in different total weights. This will again have an impact on model elements that vary with weight, such as the energy consumption of a UAV.

The thesis minimizes four objectives; flight time, risk, visual pollution and noise pollution, while ensuring the energy consumption of each UAV does not violate a capacity constraint. Thus, we address the gap of modeling visual and noise pollution. By developing quantitative measures for the visual and noise pollution generated by a UAV flying through an urban environment, more socially acceptable paths can be found. Furthermore, this thesis models *both* the ground risk and the air risk to humans associated with UAVs, which is shown to be sparse or lacking among the papers studied. This thesis also aims to provide a more realistic energy consumption model, by considering more aspects than just distance or time in the constraint. Thus, the joint consideration of visual pollution, noise pollution, air and ground risk and flight time as objectives, along with more realistic energy consumption constraints, is a novel proposition. Furthermore, though not entirely new, this thesis also considers a three dimensional and urban environment, which is less commonly studied.

Finally, this thesis uses a hybrid evolutionary algorithm consisting of NSGA-III with ACO as a subroutine to solve the MUAVPP problem. EAs have been found to be quite popular among the reviewed papers, but only Yang and Yoo (2018) use a similar hybrid evolutionary algorithm consisting of a GA and ACO. Furthermore, the use of NSGA-III is only found in the paper by Bai et al. (2022). It is also worth mentioning that Yang and Yoo (2018) only consider a single-vehicle problem, while Bai et al. (2022) consider a rural environment.

Table 3.2: Overview of reviewed literature and our thesis.

Paper	**Problem**		**Objective functions**				**Environment**				**Vehicle**		**Solution method**
	Cat.	Type	Time/ Dist.	Risk	Visual/ Noise	Other	Dims.	Static/ Dynamic	Urban/ Rural	Model	Type	Single/ Multi	
Alexopoulos and Griffin (1992), model 1	P	G	✓	-	-	-	2D	Static	-	Node net.	Robot	Single	PSE
Alexopoulos and Griffin (1992), model 2	P	G	✓	-	-	-	2D	Dynamic	-	Node net.	Robot	Single	PSH
Bellingham et al. (2003)	T	U	✓	-	-	-	2D	Dynamic	Rural	Waypoints	UAV	Multi	MP, PSH
Soltani and Fernando (2004)	P	G	✓	✓	-	✓	2D	Static	-	Grid	Vehicle*	Single	FL, PSE
Woodley and Acar (2004)	P	G	✓	-	-	-	2D	Dynamic	-	Vector	Truck	Single	ML
Hentschel et al. (2007)	T	G	-	-	-	✓	3D	Static	-	Waypoints	Truck	Single	PSE
Lamont et al. (2007)	P, R, S	V	-	✓	-	✓	3D	Static	Rural	Grid	UAV	Multi	EA
Liu et al. (2009)	R	V	✓	-	-	-	2D	Static	Rural	Node net.	Vehicle*	Multi	EA
Li et al. (2011)	P	G	✓	-	-	-	2D	Dynamic	-	Grid	Robot	Single	PSH
Tezcaner and Köksalan (2011)	P, R	V	✓	-	-	✓	2D	Static	Rural	Grid	UAV	Single	PSE
Chen, Han et al. (2012)	T	P	-	-	-	✓	3D	Dynamic	Rural	Waypoints	UAV	Single	MP
Zhang, Chen et al. (2012)	R, T	V	-	-	-	✓	3D	Static	Rural	Node net.	UAV	Single	PSH
Di Franco and Buttazzo (2015)	P	V	-	-	-	✓	2D	Static	Rural	Node net.	UAV	Single	PSH
Fügenschuh and Müllenstedt (2015)	R, T	V	✓	-	-	✓	3D	Static	Rural	Vector	UAV	Multi	MP
Yang, Tang et al. (2015)	P	G	✓	✓	-	✓	3D	Static	Rural	Waypoints	UAV	Single	EA
Yao and Zhao (2015)	P	V	-	-	-	✓	2D	Dynamic	Rural	Vector	UAV	Single	MPC
Chen, Xu et al. (2017)	R	V	✓	-	-	-	2D	Static	Rural	Node net.	UAV	Multi	EA
Hayat et al. (2017)	P	I	✓	-	-	-	2D	Static	Rural	Grid	UAV	Multi	EA
Hu, Naeem et al. (2017)	P	G	✓	✓	-	✓	2D	Dynamic	-	Waypoints	Ship	Single	EA
Modares et al. (2017)	P	I	-	-	-	✓	2D	Static	Rural	Waypoints	UAV	Multi	PSH
Ma et al. (2018)	P	G	✓	✓	-	✓	2D	Static	-	Vector	Ship	Single	EA
Wu et al. (2018)	P	I	✓	✓	-	✓	3D	Static	Urban	Vector	UAV	Single	EA
Yang and Yoo (2018)	P	I	✓	✓	-	✓	3D	Dynamic	Urban	Grid	UAV	Single	EA
Yin et al. (2018)	P	U	✓	✓	-	-	2D	Dynamic	Urban	Grid	UAV	Single	PSH
Zaccone and Martelli (2018)	P	G	✓	✓	-	-	2D	Dynamic	-	Vector	Ship	Single	ML
Ren et al. (2019)	P	G	✓	✓	-	-	3D	Static	Urban	Grid	UAV	Single	EA
Baloch and Gzara (2020)	F	V	-	-	-	✓	2D	Static	Urban	Node net.	UAV	Multi	BD
Chen, Huang et al. (2020)	P	V	✓	-	-	-	2D	Static	-	Grid	Ship	Single	ML, PSE
Dasdemir et al. (2020)	R, T	V	✓	-	-	✓	2D	Static	Rural	Vector	UAV	Single	EA
Murray and Raj (2020)	R	V	✓	-	-	-	2D	Static	Urban	Node net.	UAV & Truck	Both	MP, PSH
Shao et al. (2020)	P	G	✓	✓	-	✓	3D	Static	Rural	Vector	UAV	Multi	EA
Shivgan and Dong (2020)	P, R	I	-	-	-	✓	2D	Static	Rural	Waypoints	UAV	Single	EA
Chen, Du et al. (2021)	P	I	✓	-	-	-	2D	Static	Rural	Node net.	UAV	Multi	MP, PSH
Peng et al. (2022)	P	V	-	✓	-	✓	3D	Static	Urban	Node net.	UAV	Single	EA
Sedov, Polishchuk and Vishwanath (2021)	P	G	✓	✓	-	-	2D	Static	Urban	Grid	UAV	Single	DP
Qadir et al. (2022)	P	I	-	-	-	✓	2D	Both	Both	Waypoints	UAV	Single	EA
Bai et al. (2022)	T	P	✓	✓	-	✓	3D	Static	Rural	Waypoints	UAV	Multi	EA
Our thesis	**P**	**V**	✓	✓	✓	-	**3D**	**Static**	**Urban**	**Grid**	**UAV**	**Multi**	**EA**

In the problem category column "Cat." we have P: Path planning, R: Route planning, T: Trajectory planning, F: Facility location, S: Swarm behavior. In the problem type column "Type" we have V: Visit, I: Information gathering, G: Get to destination, P: Target pursuit, U: Unspecified. In the "Solution method" column, we have DP: Dynamic programming, EA: Evolutionary algorithm, PSE: Problem-specific exact method, MP: Mathematical programming (exact), PSH: Problem-specific heuristic, MPC: Model predictive control algorithm, BD: Benders' decomposition, FL: Fuzzy logic, ML: Machine Learning.

*Unspecified vehicle type

Chapter 4

Problem Description

The problem studied in this thesis is the static, multi-UAV path planning (MUAVPP) problem. The aim of this problem is to develop flight paths with associated timestamps for a system of UAVs, which we refer to as a *system flight plan*. The generation of a system flight plan consists of deciding what paths the UAVs should take on their respective *voyages*. Each voyage has a given start time and consists of a set of given locations that must be visited in a given order. Thus, a *path* for a UAV consists of a *segment path* on each segment between locations on the UAV's voyage. At each location, the UAV descends to the ground, performs a service and then flies back up and continues its voyage. An illustration of a system flight plan is shown in Figure 4.1. The environment is static and hence does not change between path planning and actual flight. All path planning decisions are thus made before the UAVs take off.

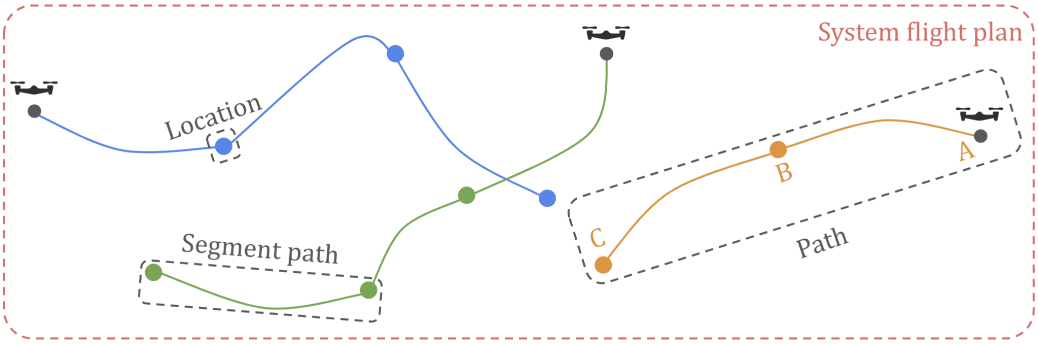


Figure 4.1: Illustration of a system flight plan for a system of three UAVs. The gray dots represent starting points for each UAV, and the path of each UAV is shown in a different color. The orange path is an implementation of the voyage consisting of visiting locations A, B and then C. This voyage thus consists of two segments; A to B and B to C. The path and associated timestamps for each of the three UAVs constitutes the system flight plan.

The locations visited by the UAVs during their voyages are all within a bounded geographical area, more precisely a three dimensional urban environment. When encountering an obstacle, a UAV can thus either fly around the obstacle while keeping the same altitude or change the altitude and fly above or below the obstacle. Furthermore, other properties of the urban environment, such as the population densities and landscape characteristics, e.g. parks or market squares, are also given.

To determine a path for each UAV, there is a trade-off between four factors: flight time, risk, visual pollution and noise pollution. Thus, the MUAVPP problem is a multi-criteria problem. Each UAV voyage has specific characteristics that are affected by the UAV type and use case, which impact how important the different criteria are when finding a path. The *importance of efficiency* characteristic denotes whether a short flight time is of particular significance. A short flight time is essential when e.g. an emergency medical UAV transports a defibrillator to an individual in danger. The importance of efficiency is given for each segment for every UAV in the system, and it may vary between segments belonging to the same UAV's voyage. Each UAV must take the paths of other UAVs into account to avoid collisions. A UAV is not allowed to wait by hovering in the air, and its path must thus be adjusted if it gets too close to another UAV at any point in time. Furthermore, a UAV may not wait on the ground after performing a service, unless ascent results in an immediate collision.

Each UAV has a set of given characteristics, including its weight, size and speed, which impact what paths the UAV can take. Each UAV also has a limited energy capacity which cannot be exceeded. The UAVs fly with constant speed. We assume that the UAVs operate during the daytime and that the weather is stable. Regulations also impact the paths the UAVs can take. In particular, altitude regulations and no-flight zones must be taken into account. The altitude regulations entail that the UAVs must fly between a given maximum and minimum altitude above ground.

The goal of the MUAVPP problem is to find a set of optimal system flight plans by balancing the flight time, risk, visual pollution and noise pollution criteria for the entire system of UAVs. The importance of the different criteria depends on the characteristics of each of the UAVs and the segments along their respective voyages.

Chapter 5

Mathematical Model

This chapter presents a mathematical model of the static, multi-UAV path planning (MUAVPP) problem described in Chapter 4. The model is a mixed integer nonlinear program (MINLP). Section 5.1 describes the mathematical model and the underlying assumptions. Section 5.2 presents the sets, parameters and decision variables used in the model. The objective functions are explained in Section 5.3 and the constraints in Section 5.4. The chapter uses a compact formulation of the models for energy consumption, risk, visual pollution and noise pollution. A full overview of the equations for these models is given in Chapter 6 and summarized in Appendix B.

5.1 Model Description and Assumptions

The static MUAVPP problem is modeled as a multi-objective optimization problem with the set of objectives being flight time, risk, visual pollution and noise pollution. Despite considering a system of UAVs, many of the UAV constraints can be handled independently in the model. There are two aspects connecting the UAVs in the model, however. The first is the collision avoidance requirement. The second is the fact that UAVs operating in close proximity can affect the total amount of risk, visual pollution and noise pollution generated in a nonlinear way. This effect can potentially make it beneficial to ensure that the UAVs either fly close or keep their distance from one another. These interactive effects are explained in Chapter 6.

There are a number of assumptions made about the input to the problem. The first is the discretization of the environment into a grid, and the definition of a graph of edges and vertices on this grid. The set of vertices is assumed to be preprocessed, so that only feasible vertices are included. This means that vertices positioned in no-flight zones, within obstacles or outside the minimum and maximum allowed flight altitudes above ground, are removed. This set of assumptions results in a rather compact mathematical model in terms of constraints, as the grid limitations are mostly taken care of through the preprocessing. Locations that must be visited by each UAV correspond to vertices in the graph, and consecutive locations to visit thus represent start and end vertices of segments. We assume that all locations the UAVs must visit are at the minimum allowed flight altitude above ground. When a UAV reaches the end vertex of a segment, it flies down to the ground, performs a service and flies back up to the same vertex. Thus, the end vertex of each segment is also the start vertex of the next segment. Finally, we assume that the flight time and energy consumption of each UAV as it moves between vertices in the graph are given.

5.2 Definition of Sets, Parameters and Decision Variables

This section presents the sets, parameters and decision variables used in the MINLP model.

5.2.1 Sets

$\mathcal{N}$	Set of vertices
$\mathcal{N}_i$	Set of neighboring vertices to vertex i, $\mathcal{N}_i \subseteq \mathcal{N}$
$\mathcal{N}_i^H$	Set of horizontal neighboring vertices to vertex i, $\mathcal{N}_i^H \subseteq \mathcal{N}$
$\mathcal{N}_{jl}^Q$	Set of pairs of vertices $[i, k]$ that create horizontal squares with vertices j and l as end vertices on diagonal edges (i, j) and (k, l)
$\mathcal{U}$	Set of UAVs
$\mathcal{S}_u$	Set of segments included in the voyage for UAV u

5.2.2 Parameters

E_u^{Max}	Energy capacity of UAV u
E_{uijs}	Energy consumed flying from vertex i to vertex j on segment s for UAV u
I_{uis}^F	1 if vertex i is the end vertex of segment s for UAV u, 0 otherwise
I_{uis}^S	1 if vertex i is the start vertex of segment s for UAV u, 0 otherwise
J_{us}	Importance of efficiency on segment s for UAV u
$P^N(\vec{t})$	Function to calculate the total noise pollution when system flight plan $\vec{t}$ is used
$P^V(\vec{t})$	Function to calculate the total visual pollution when system flight plan $\vec{t}$ is used
$R(\vec{t})$	Function to calculate the total risk when system flight plan $\vec{t}$ is used
T_u^S	Start time of the voyage of UAV u

T^{O}_{ui} Service time of UAV u, i.e. the time spent flying down to the ground directly below vertex i, performing a service and flying back up to vertex i

T_{uij} Time UAV u spends flying from vertex i to vertex j

$\underline{T}^{D}$ Minimum time difference between two UAVs traversing intersecting diagonal edges

$\underline{T}^{E}$ Minimum time difference between two UAVs traversing the same edge in opposite directions

$\underline{T}^{V}$ Minimum time difference between two UAVs visiting the same vertex

5.2.3 Decision Variables

t_{uis} Time when UAV u enters vertex i on segment s, vectorized as $\vec{\boldsymbol{t}}$

$$x_{uijs} = \begin{cases} 1, & \text{UAV } u \text{ flies directly from vertex } i \text{ to vertex } j \text{ on segment } s \\ 0, & \text{Otherwise} \end{cases}$$

5.3 Objective Functions

The objective function vector (5.1) consists of four objective functions which are all to be minimized. The first objective function is to minimize the total flight time. T_{uij} is a generalized time parameter, and J_{us} is a weighting parameter for each segment and UAV. J_{us} is used to indicate the importance of efficiency on each segment of a UAV's voyage. Since the service time at locations is constant, it is not included in the objective value for flight time. The second objective is to minimize the total risk, i.e. the combined air and ground risk. The third and fourth objectives are to minimize the total visual pollution and noise pollution, given as $P^{V}(\vec{\boldsymbol{t}})$ and $P^{N}(\vec{\boldsymbol{t}})$, respectively. The $R(\vec{\boldsymbol{t}})$, $P^{V}(\vec{\boldsymbol{t}})$ and $P^{N}(\vec{\boldsymbol{t}})$ functions are explained in-depth in Chapter 6. While the flight time objective is calculated for each separate UAV based on traversed edges, the remaining three objectives are calculated based on entire system flight plans $\vec{\boldsymbol{t}}$. A system flight plan consists of the set of t_{uis} variables, i.e. a system flight plan is defined as a set of variables stating which UAVs are in which vertices at what time.

$$min \quad \left[\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} J_{us} \cdot T_{uij} \cdot x_{uijs} \ , \ R(\vec{\boldsymbol{t}}) \ , \ P^{V}(\vec{\boldsymbol{t}}) \ , \ P^{N}(\vec{\boldsymbol{t}}) \right] \tag{5.1}$$

5.4 Constraints

This section presents the four types of constraints in the MINLP model; flow constraints, time constraints, collision avoidance constraints and energy consumption constraints.

5.4.1 Flow Constraints

The flow constraints ensure that the MINLP model produces a single, connected path both within each segment and across the different segments of each UAV's voyage. Constraints (5.2) state that for all vertices that are not the start or end vertex on a given segment s for UAV u, the number of times a vertex i is entered must equal the number of times vertex i is exited.

$$\sum_{j \in \mathcal{N}_i} x_{uijs} - \sum_{k \in \mathcal{N}_i} x_{ukis} = 0 \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \mid I^S_{uis} + I^F_{uis} = 0 \quad (5.2)$$

For the start and end vertex on each segment, the flow constraints are slightly altered. Since all objectives are positive, non-decreasing and to be minimized, multiple visits to the start and end vertices of a segment would only worsen the objective values. Thus, constraints (5.3) limit the start vertex i to only be *exited* once. Conversely, constraints (5.4) limit the end vertex i to only be *entered* once.

$$\sum_{j \in \mathcal{N}_i} x_{uijs} = 1 \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \mid I^S_{uis} = 1 \quad (5.3)$$

$$\sum_{j \in \mathcal{N}_i} x_{ujis} = 1 \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \mid I^F_{uis} = 1 \quad (5.4)$$

Finally, constraints (5.5) require that each UAV visits a given vertex at most once on each segment. These constraints ensure that the UAVs cannot indirectly "wait" on a segment path, by moving back and forth between two or more vertices.

$$\sum_{j \in \mathcal{N}_i} x_{ujis} \leq 1 \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \quad (5.5)$$

5.4.2 Time Constraints

The time constraints ensure a cohesive flow of time in the MINLP model. The nonlinear constraints (5.6) require the point in time when UAV u is positioned in vertex j on segment s to equal the point in time when the UAV was residing in a neighboring vertex i plus the time needed to fly from i to j. Constraints (5.7) ensure that every UAV starts its flight at its specified start time T^S_u. Constraints (5.8) set the time t_{uis} of the start vertex of each intermediate segment of a UAV. The earliest time for t_{uis} is the time the UAV reached the same vertex as the end of the previous segment, plus the service time T^O_{ui}. To allow the UAV to avoid a collision immediately upon ascent, t_{uis} may be further delayed by up to the largest minimum time difference to avoid collision.

$$t_{ujs} = \sum_{i \in \mathcal{N}_j} (t_{uis} + T_{uij}) \cdot x_{uijs} \qquad u \in \mathcal{U}, j \in \mathcal{N}, s \in \mathcal{S}_u \quad (5.6)$$

$$t_{ui1} = T_u^S \qquad u \in \mathcal{U}, i \in \mathcal{N} \mid I_{ui1}^S = 1 \quad (5.7)$$

$$0 \leq t_{ui(s+1)} - (t_{uis} + T_{ui}^O) \leq \max(\underline{T}^D, \underline{T}^E, \underline{T}^V) \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \setminus \{|\mathcal{S}_u|\} \mid I_{uis}^F = 1 \quad (5.8)$$

5.4.3 Collision Avoidance Constraints

Collision avoidance constraints are added to the MINLP model since each UAV must consider the paths taken by the other UAVs in the system. The nonlinear constraints (5.9), (5.10) and (5.11) each cover one of the three possible collision types, shown in Figure 5.1.

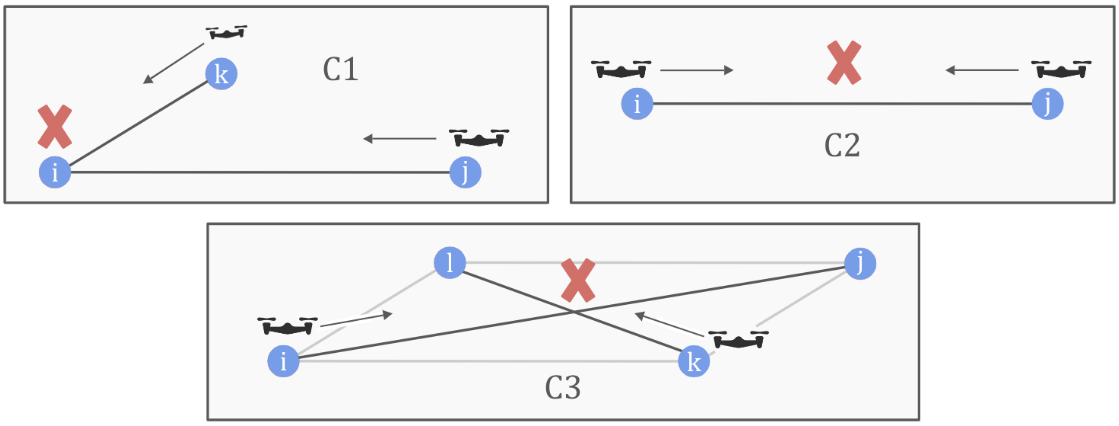


Figure 5.1: Illustration of possible UAV collision types, C1-C3. Collision type C1 is shown at the top left, where two UAVs want to fly to the same vertex i at similar points in time. Collision type C2 is shown at the top right, where two UAVs want to fly across the same edge in opposite directions at similar points in time. Collision type C3 is shown at the bottom, where two UAVs want to fly along intersecting diagonals at similar points in time, with one UAV wanting to fly from vertex i to j and the other UAV from vertex k to l. C3 also exemplifies the N_{jl}^Q set, which consists of the two remaining vertices $[i, k]$ needed to create a horizontal square with vertices j and l as end vertices on diagonal edges (i, j) and (k, l).

Constraints (5.9) avoid collision type C1 by stating that two UAVs cannot enter the same vertex with a time difference less than $\underline{T}^V$. Constraints (5.10) avoid collision type C2 by ensuring that two UAVs cannot traverse the same edge in opposite directions with a time difference less than $\underline{T}^E$. Finally, constraints (5.11) avoid collision type C3 by prohibiting two UAVs to cross intersecting diagonal edges with a time difference less than $\underline{T}^D$. The set N_{jl}^Q is also exemplified in the graph at the bottom part of Figure 5.1.

$$|t_{uis} - t_{u'is'}| \geq \underline{T}^V \cdot \left(\sum_{j \in \mathcal{N}_i} x_{ujis} + \sum_{k \in \mathcal{N}_i} x_{u'kis'} - 1 \right) \qquad u,u' \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u, s' \in \mathcal{S}_{u'} \mid u \neq u' \tag{5.9}$$

$$|t_{uis} \cdot x_{ujis} - t_{u'js'} \cdot x_{u'ijs'}| \geq \underline{T}^E \qquad u,u' \in \mathcal{U}, i \in \mathcal{N}, j \in \mathcal{N}_i, s \in \mathcal{S}_u, s' \in \mathcal{S}_{u'} \mid u \neq u' \tag{5.10}$$

$$|t_{ujs} \cdot x_{uijs} - t_{u'ls'} \cdot x_{u'kls'}| \geq \underline{T}^D \qquad u,u' \in \mathcal{U}, j \in \mathcal{N}, l \in \mathcal{N}_j^H, \; [i,k] \in \mathcal{N}_{jl}^Q, s \in \mathcal{S}_u, s' \in \mathcal{S}_{u'} \mid u \neq u' \tag{5.11}$$

5.4.4 Energy Consumption Constraints

The energy consumption constraints (5.12) ensure that each UAV does not consume more energy than E_u^{Max}. The equation for calculating E_{uijs} is explained in Chapter 6.

$$\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} E_{uijs} \cdot x_{uijs} \leq E_u^{Max} \qquad u \in \mathcal{U} \tag{5.12}$$

5.4.5 Binary and Non-Negativity Constraints

Naturally, the variables have non-negativity constraints, covered by constraints (5.13) and the binary constraints (5.14).

$$t_{uis} \geq 0 \qquad u \in \mathcal{U}, i \in \mathcal{N}, s \in \mathcal{S}_u \tag{5.13}$$

$$x_{uijs} \in \{0,1\} \qquad u \in \mathcal{U}, i \in \mathcal{N}, j \in \mathcal{N}_i, s \in \mathcal{S}_u \tag{5.14}$$

Chapter 6

World and Objective Models

This chapter presents the models for the environment, energy consumption and the objectives used in this thesis. Section 6.1 describes the environment model and how energy consumption and the various objectives are connected to this model. Section 6.2 explains the model of energy consumption. Both the models for the environment and energy consumption are revised and updated versions of the models used in our preparatory research project. Sections 6.3, 6.4 and 6.5 present the models for the risk, visual pollution and noise pollution objectives, respectively. For each of the three objective model sections, we first propose the model for calculating the objective from a single-UAV perspective, which is based on work done in our preparatory research project. We then present the necessary updates to take the system perspective into account. A compact representation of the models for energy consumption and the objectives is given in Appendix B. Note that interaction between the objectives, e.g. that noise pollution from a UAV may be perceived worse if an individual sees the UAV while hearing it, is not taken into account. All the equations displayed in this chapter are developed with an urban environment in mind.

6.1 Environment Model

The static MUAVPP problem explained in Chapter 4 takes place in a discrete, three dimensional urban environment. The environment is defined by a bounded map of the landscape, and is discretized into a grid. The length of a gridline between gridline intersections is denoted L. An undirected, weighted graph with a set of vertices and edges is defined on the grid. The edges represent where the UAVs can fly, while the vertices represent either locations that have to be visited by a UAV or possible traversal points on a voyage. The UAVs are allowed to fly vertically upwards or downwards, or horizontally right, left, forwards, backwards or diagonally. Hence, each vertex, apart from the vertices in the outermost part of the graph, has ten neighbors, as shown in the left and middle part of Figure 6.1.

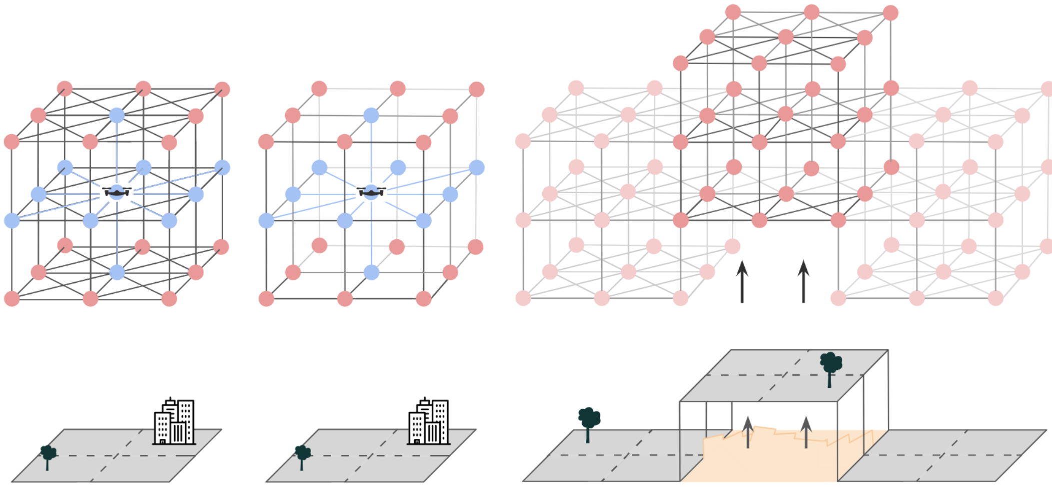


Figure 6.1: In the left and middle part of the figure the blue vertices and edges are feasible moves for a UAV in a given vertex. The middle part of the figure has a simplified graph compared to the left part to improve readability. The gray squares at the bottom are ground squares, which are separate from the flyable graph. The ground square multiple λ is set equal to 2 for all parts of the figure, thus leading to one ground square below the left and middle parts of the figure, and three ground squares below the right part. The dashed lines illustrate the projection of the graph onto the ground squares, omitting the diagonal edges for readability. The right part of the figure shows how the graph shifts based on the altitude of a ground square.

The initial graph must be modified. First, vertices that interfere with static obstacles and no-flight zones are removed as they cannot be used by UAVs. Vertices outside the minimum and maximum allowed flight altitudes above ground are also removed, thus dividing the environment into a flyable graph in the air and a separate set of *ground squares* on the ground, as shown in Figure 6.1. We hence define $\mathcal{G}$ to be the set of all ground squares, indexed by g. The ground squares do not need to be as fine-grained as the gridline sections in the grid, and the side length of a ground square is therefore set to a multiple λ of L. The altitude above sea of each ground square is rounded up to the closest multiple of L. Thus, the flyable graph shifts upwards or downwards based on the altitude of a ground square, as shown in the left part of Figure 6.1, to ensure that the UAVs always stay within the allowed flight altitudes above ground. We assume that the minimum allowed flight altitude is strictly greater than zero. Due to these modifications, some of the vertices that are not in the outermost part of the graph might also end up with less than ten neighbors.

The objectives and energy consumption of a UAV u are associated with various parts of the graph. Each vertex i has an associated measure of visual and noise pollution, denoted by P^V_{ui}

and P^N_{ui}, respectively. The weight of an edge (i, j) on a segment s is a vector $(J_{us} \cdot T_{uij}, R_{uijs}, E_{uijs})$, representing the importance of efficiency-adjusted flight time, the risk and the energy consumption, respectively. These relationships are displayed in Figure 6.2. The reason for this allocation is that the equations for visual and noise pollution do not incorporate time, while energy consumption and risk do. The total flight time and energy consumption of a given UAV is thus found by simply summing over all edges traversed in the UAV's path. The total objective values for risk, visual pollution and noise pollution require more sophisticated methods of summation when considering a system of UAVs, and are explained in the following sections.

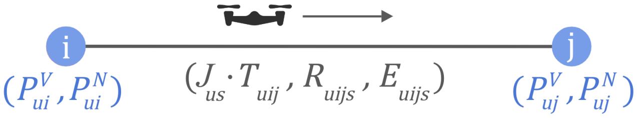


Figure 6.2: Illustration of measures associated with vertices and edges in the graph. P^V_{ui} and P^N_{ui} are the values of visual pollution and noise pollution associated with vertex i for UAV u. The edge weight vector consists of the importance of efficiency-adjusted flight time T_{uij}, the risk R_{uijs} and the energy consumption E_{uijs} associated with traversing the edge between vertices i an j for UAV u on segment s.

6.2 Energy Consumption Model

Most energy consumption models for path planning problems include time or distance limitations, as described in Chapter 3, with some models further assuming constant energy consumption per unit of time or distance traveled. A basic model of energy consumption is stated as the energy consumed per unit of time, multiplied by the time (Yang & Yoo, 2018). To give a more realistic view of energy consumption, this thesis extends this simple energy consumption model by taking the effects of payloads, flight altitude and flight direction, i.e. whether the UAV is flying horizontally, vertically upwards or vertically downwards, into account. Taking inspiration from Majeed et al. (2020), we thus end up with the following model for energy consumption for a given UAV u on an edge going from vertex i to j on segment s

$$E_{uijs} = T_{uij} \cdot (W_u + W^P_{us}) \cdot (E^W_u + E^F_{ui}), \quad \text{where } E^F_{ui} = \begin{cases} H^S_i \cdot E^A_u, & \text{if } j \text{ is horizontal to } i \\ H^S_i \cdot E^U_u, & \text{if } j \text{ is above } i \\ H^S_i \cdot E^D_u, & \text{if } j \text{ is below } i \end{cases} \tag{6.1}$$

In equation (6.1), E^W_u is the energy consumed per kilogram per unit of time UAV u is in active mode, i.e. not switched off. W_u is the weight of UAV u and W^P_{us} is the weight of the payload carried by UAV u on segment s, which may potentially be zero. The effect of flight direction on energy consumption, E^F_{ui}, is split into three cases. E^A_u, E^U_u and E^D_u denote energy consumed traveling horizontally, vertically upwards and vertically downwards, respectively, per kilogram of weight per meter above sea level per unit of time, with $E_u^U > E_u^D$. H_i^S is the altitude above sea level of vertex i of edge (i, j). Note that this thesis ignores the energy consumed by a UAV during service, i.e. when it descends to the ground from a vertex corresponding to a location that must be visited, performs a service on the ground and flies back up to the same vertex.

6.3 Risk Model

A common way to measure the risk associated with UAVs is a probabilistic approach where the expected fatality rate (EFR) is calculated, as described in Section 2.5. The state of the art in the field only considers the risk for humans, excluding the risk to animals and buildings, and we make the same simplification. However, the risk to animals and protected buildings may still be partly taken into consideration through the use of no-flight zones in e.g. nature conservation areas. The EFR associated with a UAV is calculated as follows

$$EFR = \frac{N^{Exp} \cdot B}{T^F} \tag{6.2}$$

where T^F is the mean time between failures for the UAV, N^{Exp} is the expected number of individuals at risk when the UAV falls down and B is the probability of a fatality, given that an individual is at risk (Sedov, Polishchuk & Vishwanath, 2021).

The risk associated with UAVs must be calculated from two different perspectives. The first is from the perspective of a single, isolated UAV. The second perspective is for the entire *system* of UAVs, i.e. how much risk a system flight plan generates, taking the simultaneous risk effects from multiple UAVs into account. This section first develops a model to calculate the risk value from the single-UAV perspective and then from the system perspective.

6.3.1 Single Unmanned Aerial Vehicle Perspective

To quantify the risk associated with UAVs, this thesis uses *expected fatalities*, given as EFR multiplied by flight time. The total expected fatalities is the sum of the expected fatalities associated with air risk and the expected fatalities associated with ground risk. The total risk associated with a given path for a single UAV u is given as

$$\begin{aligned}
R_u &= \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} \frac{1}{T_u^F} \cdot (N_{us}^A \cdot B^A + N_{uijs}^G \cdot B_{uijs}^G) \cdot T_{uij} \cdot x_{uijs} \,, \\
\text{where} \quad & N_{us}^A = K_{us} \,, \quad B^A = 1 \,, \quad N_{uijs}^G = (W_u + W_{us}^P) \cdot \left(M_u^L + \frac{H^P}{sin(A_u^G)} \right) \cdot \frac{1}{|\mathcal{G}_{uijs}^R|} \sum_{g \in \mathcal{G}_{uijs}^R} D_g \,, \\
B_{uijs}^G &= \frac{1}{|\mathcal{G}_{uijs}^R|} \sum_{g \in \mathcal{G}_{uijs}^R} \frac{1}{1 + \sqrt{\frac{C^{R,Mid}}{C^{R,Low}}} \cdot \left(\frac{C^{R,Low}}{\frac{1}{2} \cdot (W_u + W_{us}^P) \cdot (C^G \cdot (H_i^G + H_j^G) + (V_u)^2)} \right)^{\frac{1}{4F_g}}}
\end{aligned} \tag{6.3}$$

In equation (6.3), T_u^F is the mean time between failures for UAV u. T_u^F only depends on UAV characteristics, and is therefore constant for a given UAV. N_{us}^A and N_{uijs}^G are the expected number of individuals at risk associated with air and ground risk, respectively. B^A and B_{uijs}^G are the probabilities of fatality given that an individual is at risk, for air and ground risk, respectively.

Air Risk Model

Equation (6.3) only considers the risk for the passengers aboard UAV u when calculating the air risk. This is because the air risk associated with UAVs for the passengers on board other aerial vehicles than UAVs is virtually zero, as described in Subsection 2.5.1. Thus, N_{us}^A is assumed to equal the number of individuals on board UAV u on the segment s, K_{us}. B^A is set to 1, similar as for conventional aviation (Sedov, Polishchuk, Maury et al., 2021).

Ground Risk Model

The ground risk for a given UAV u on segment s varies based on the edge (i, j) traversed by the UAV. The variation is due to N_{uijs}^G and B_{uijs}^G depending on the population density and the sheltering in the area underneath the UAV. In this thesis, N_{uijs}^G and B_{uijs}^G are calculated assuming that UAV u is located at the middle of the edge between vertex i and vertex j.

Equation (6.3) estimates the expected number of individuals on the ground at risk, N_{uijs}^G, as the *expected number of individuals* in the *lethal area* of a UAV (Dalamagkidis et al., 2008; Sedov, Polishchuk & Vishwanath, 2021; Weibel & Hansman, 2015). The lethal area is the area at risk of being hit by the UAV and potential debris if the UAV falls down, and is given in the first two factors of N_{uijs}^G in equation (6.3). W_u and W_{us}^P are the weights of the UAV and its payload, respectively, M_u^L is the length of the UAV and H^P is the average height of a person. A_u^G is the *glide angle*, i.e. the angle of the UAV's descent. The calculation of the lethal area was proposed by Dalamagkidis et al. (2008) and the observant reader may notice that the unit for the resulting value is not m^2. However, the resulting value corresponds well with the size of the lethal area, and is therefore used as an estimation. The lethal area is assumed to be a circle in this thesis. The *expected number of individuals* in the lethal area is estimated using the average of the population densities, D_g, of the ground squares within the lethal area and constitutes the third factor of N_{uijs}^G in equation (6.3). $\mathcal{G}_{uijs}^R$ is the set of ground squares fully included in the lethal area when UAV u flies between vertex i and vertex j on segment s. If no ground squares are fully included, the ground squares that are covered *the most* are included in $\mathcal{G}_{uijs}^R$. Figure 6.3 illustrates two lethal areas and the ground squares included in each of them.

Equation (6.3) estimates the probability of fatality, B_{uijs}^G, as the average probability of fatality for the ground squares included in $\mathcal{G}_{uijs}^R$. The equation for the probability of fatality used for each ground square was proposed by Dalamagkidis et al. (2008). $F_g \in (0, 1]$ is the *sheltering factor* and quantifies the degree of shelter given to individuals located in a ground square g

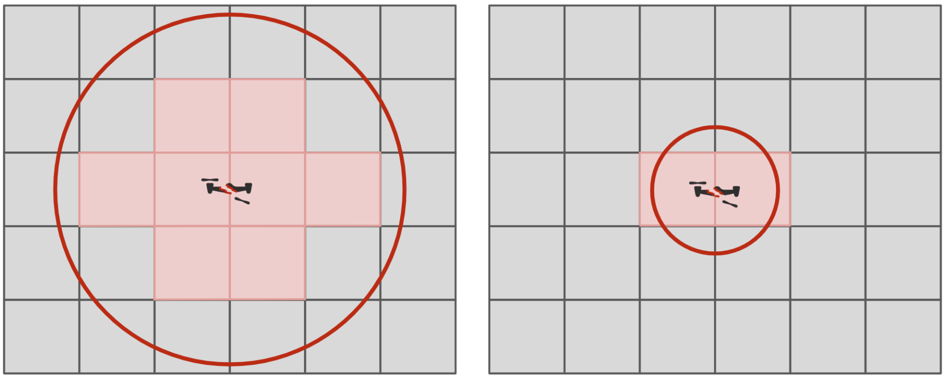


Figure 6.3: Illustration of the lethal area of a UAV. The red circle illustrates the lethal area, while the ground squares shaded in light red are the ones included in the risk calculations. In the left part of the figure, multiple ground squares are fully included in the lethal area. No ground squares are fully included in the right part.

by e.g. buildings and trees. The sheltering factor may also include to what degree individuals on the ground are paying attention to what is happening around them, and therefore to what degree they are able to protect themselves from a falling UAV (Dalamagkidis et al., 2008; Primatesta et al., 2020). $C^{R,Mid}$ is the energy required at impact for a fatality probability of 50%, while $C^{R,Low}$ is the energy required at impact to cause a fatality as F_g goes to zero (Dalamagkidis et al., 2008). The denominator below $C^{R,Low}$ is the kinetic energy at impact, assuming free fall of the UAV from a given altitude. H_i^G is the altitude above ground for vertex i, C^G is the acceleration of free fall and V_u is the speed of UAV u. This is a conservative approach, due to the free fall assumption.

6.3.2 System Perspective

When considering the risk of a system flight plan, the risk of collisions must be incorporated. The time difference requirements of constraints (5.9) through (5.11) in the MINLP model in Chapter 5 may equivalently be stated as distance requirements between UAVs. While the MINLP model avoids guaranteed collisions through the collision avoidance constraints, the risk of collisions is still present if multiple UAVs are in close proximity to each other. To handle the increased risk associated with a system of UAVs, time is discretized into a set of time intervals with length δ, and D^{Min} and D^{Max} distances are given. The described situation is shown in Figure 6.4. The circle shaded in light red with radius D^{Min} is called the *collision area*, and no UAVs may fly closer than this to any other UAV. D^{Min} must thus equal the distance equivalent of the largest time difference needed to avoid collision, to ensure the collision constraints in the MINLP model are upheld. The area shaded in light blue and bounded by D^{Max} in Figure 6.4 is called the *collision risk area*. UAVs may fly within the collision risk areas of other UAVs, but it increases the risk value compared to when there are no other UAVs in the collision risk area within a given time interval. We hence refer to D^{Max} as the *collision risk threshold*.

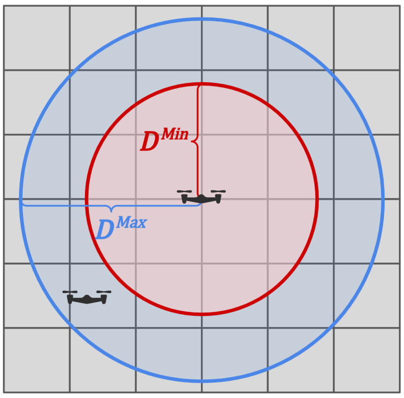


Figure 6.4: Illustration of the collision area and the collision risk area. The collision area is the area shaded in light red, and the collision risk area is the area shaded in light blue. D^{Min} is the minimum distance between two UAVs to avoid collision. The collision risk threshold, D^{Max}, is the maximum distance between two UAVs that results in increased risks.

The risk associated with a given system flight plan $\vec{\boldsymbol{t}}$ is given as

$$R(\vec{\boldsymbol{t}}) = \sum_{\tau \in \mathcal{T}} \sum_{u \in \mathcal{U}} \sum_{\substack{i \in \\ \mathcal{N}(u,s,\tau,\delta,\vec{\boldsymbol{t}})}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} \frac{1}{T_u^F - T^C(u,\tau,\delta,\vec{\boldsymbol{t}})} \cdot (N_{us}^A \cdot B^A + N_{uijs}^G \cdot B_{uijs}^G) \cdot T_{uij} \cdot x_{uijs} ,$$

$$\text{where } T^C(u,\tau,\delta,\vec{\boldsymbol{t}}) = \alpha \cdot T_u^F \cdot min\left(\sum_{\substack{u' \in \\ \mathcal{U}^C(u,\tau,\delta,\vec{\boldsymbol{t}})}} \frac{\frac{W_{u'}}{W_u^{Max}}}{\min(D(u,u',\tau,\vec{\boldsymbol{t}}), D(u,u',\tau+\delta,\vec{\boldsymbol{t}})) - D^{Min} + 1}, 1 \right) \tag{6.4}$$

In equation (6.4), $\mathcal{T}$ is the set of start times, τ, for time intervals of length δ. $\mathcal{N}(u,\tau,\delta,s,\vec{\boldsymbol{t}})$ is the set of vertices visited by UAV u on segment s in system flight plan $\vec{\boldsymbol{t}}$ in time interval $[\tau, \tau+\delta)$. The system perspective model is almost equal to the single-UAV perspective model shown in equation (6.3), using the same equations for N_{us}^A, B^A, N_{uijs}^G, and B_{uijs}^G. The only difference is that the denominator in the fraction which has been updated to ensure an increased risk value in the collision risk area. The denominator is no longer the mean time between failures T_u^F, but the *mean time between falldowns*. We assume that the only possible causes of a UAV falling down are inherent failures that depend on the UAV's characteristics and mid-air collisions. Thus, T_u^F is decreased using a function $T^C(u,\tau,\delta,\vec{\boldsymbol{t}})$, which depends on what UAVs reside within the collision risk area of UAV u in the time interval $[\tau, \tau+\delta)$. $T^C(u,\tau,\delta,\vec{\boldsymbol{t}})$ is assumed to be zero if no UAVs reside within the collision risk area of UAV u.

In the collision risk adjustment function $T^C(u,\tau,\delta,\vec{\boldsymbol{t}})$, $\alpha \in [0,1)$ is a factor that decides the maximum reduction of the mean time between falldowns due to the possibility of collisions. With α equal to 0, the UAVs in the system only affect each other through the collision avoidance constraints. As α increases, the system effect becomes more and more prominent by allowing for increased risk when UAVs fly within each other's collision risk areas. $\mathcal{U}^C(u, \tau, \delta, \vec{t})$ is the set of UAVs that affect the risk of mid-air collisions for UAV u, i.e. are within D^{Max} from UAV u, in time interval $[\tau, \tau + \delta)$ in system flight plan $\vec{t}$. As described in Section 3.3 in the literature review, a common measure of collision risk is the distance to the closest moving obstacle or other vehicle. Distance is incorporated as the denominator of the collision risk adjustment function, with $D(u, u', t, \vec{t})$ being the distance between UAV u and UAV u' at time t in system flight plan $\vec{t}$. There is more risk associated with flying in close proximity to a heavy UAV compared to a lightweight UAV. Therefore, the weight of each other UAV, $W_{u'}$, that are within the collision risk area are included in the $T^C(u, \tau, \delta, \vec{t})$ function. We consider the distances between UAVs to be more important than weight in regards to whether collisions occur. Hence, distance is given as an absolute value, while the part of $T^C(u, \tau, \delta, \vec{t})$ concerning UAV weight is given as a ratio, with W_u^{Max} being the baseline used to decide the importance of weight.

6.4 Visual Pollution Model

Little research exists on the quantification of visual pollution. As described in Section 2.6, a study by Thomas (2022) found that the number of UAVs observed appears to be the most important factor for visual pollution, followed by the distance to the closest UAV. The study presents the following equation quantifying the amount of visual pollution from the perspective of a single individual at a given point in time

$$P^V = a \cdot \frac{(\textit{No. of UAVs})^b}{(\textit{Distance to closest UAV})^c} + d \tag{6.5}$$

where a, b, c and d are constants. A larger value in equation (6.5) means more visual pollution. Since the equation only looks at the distance to the closest UAV, the visual pollution of e.g. one UAV at 60 meters altitude and four at 200 meters altitude will give the same visual pollution as if all UAVs were at 60 meters altitude. Due to this weakness, we only use equation (6.5) as a starting point for our model of visual pollution for a *single* UAV and create our own equation for the visual pollution generated by multiple UAVs. The quantification of visual pollution thus takes the same two perspectives as the risk model; a single-UAV perspective and a system perspective. The single-UAV perspective finds how much visual pollution a given UAV generates when it is located in each vertex of a path. The system perspective finds the simultaneous effect of the visual pollution from multiple UAVs. For both perspectives, the subjectivity of visual pollution described in Section 2.6 is disregarded.

6.4.1 Single Unmanned Aerial Vehicle Perspective

To quantify the total visual pollution generated by UAV u in a given vertex i when the system aspect is ignored, equation (6.5) is adjusted to yield the following equation

$$P_{ui}^{V} = \sum_{g \in \mathcal{G}_{ui}^{V}} (1 - F_g) \cdot (C^{V,1} \cdot \frac{1}{(D_{ig}^{U})^{C^{V,2}}}) \cdot Q_g \tag{6.6}$$

where $C^{V,1}$ and $C^{V,2}$ correspond to the constants a and c in equation (6.5), respectively. Constants b and d are omitted, since equation (6.6) is only used for a single UAV and we assume that there is no visual pollution if there are no UAVs. Equation (6.6) multiplies the visual pollution experienced by a single individual by the number of individuals in an affected ground square, Q_g, and takes the sheltering factor associated with each ground square, F_g, into account. D_{ig}^{U} is the Euclidean distance from vertex i, where the UAV is located, to the center of ground square g. $\mathcal{G}_{ui}^{V} \subseteq \mathcal{G}$ is the set of affected ground squares when UAV u is positioned in vertex i.

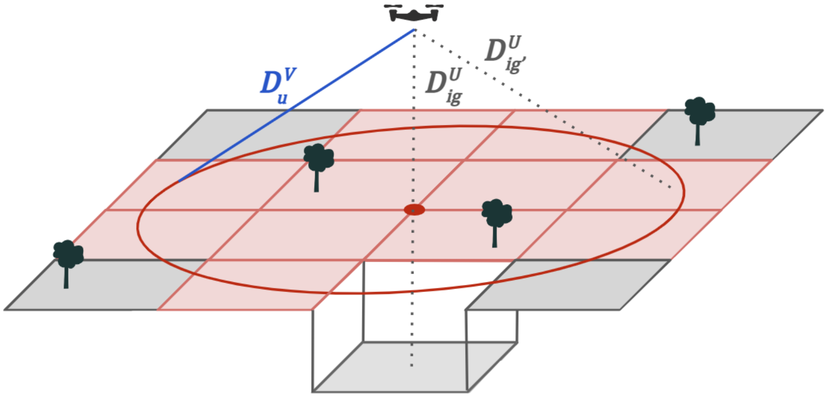


Figure 6.5: Illustration of the ground squares affected by visual pollution given a cut-off distance D_u^V. The red shaded ground squares have their center within the circle spanning the cross section of the cut-off sphere. Ground square g is not shaded, despite being within the circle, since $D_{ig}^U > D_u^V$.

To decide which ground squares are affected by visual pollution when a UAV is located in a given vertex, a threshold is used to find the cut-off distance. The cut-off distance corresponds to the radius of a sphere around a UAV, and we assume that only individuals in ground squares with centers *within* the sphere experience visual pollution from the UAV. Figure 6.5 shows how the affected ground squares may vary based on the cut-off distance and the altitude above sea of a ground square. The cut-off distance, D_u^V, is found by calculating the distance at which the length of UAV u, M_u^L, appears smaller than a given threshold, R^V, to an observer standing at the center of a ground square measuring the size of the UAV in the sky with a measurement device at a distance D^E from the observer's eyes.

$$D_u^V = \frac{M_u^L}{R^V} \cdot D^E \tag{6.7}$$

There are a few things to note about the visual pollution equation. The values from equation

(6.6) allow for linear comparisons such as a visual pollution value of 10 being twice as bad as 5, but the values have no easily interpretable meaning on their own. Furthermore, we use the value of the vertex a UAV leaves from when summing over vertices, and the visual pollution value of the final vertex on each segment is included. Thus, each intermediate location a UAV must visit between the initial and the final location on a voyage is counted twice. The double summation of these vertices intuitively makes sense, as the UAV will have to fly from each intermediate location, down to the ground to perform a service and then back up to the vertex, i.e. visit the vertex twice.

The total visual pollution of a given path for a single UAV u is given as

$$P_u^V = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} P_{ui}^V \cdot x_{uijs} + \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_u} P_{ui}^V \cdot I_{uis}^F \tag{6.8}$$

6.4.2 System Perspective

To quantify the total visual pollution generated by a system flight plan, we begin by finding the visual pollution in each ground square. Algorithm 1 calculates the total visual pollution in ground square g by dividing the entire time frame of system flight plan $\vec{\boldsymbol{t}}$ into a set of time intervals of a given length δ. For each time interval, the for-loop in lines 7-9 calculates the visual pollution values from the single-UAV perspective for each UAV that affects ground square g, i.e. each UAV that has g in its set of affected ground squares $\mathcal{G}_{ui}^V$. The equation used in line 8 is similar to equation (6.6). The only difference is that the parameter D_{ig}^U in equation (6.6) is replaced by the function $D(t_{uis}, g)$. $D(t_{uis}, g)$ finds the distance between the ground square g and the vertex i that UAV u is located in at time t_{uis}.

Lines 11-13 deal with non-linear addition of visual pollution values from multiple UAVs. The visual pollution values from the single-UAV perspective are sorted in descending order and multiplied by a factor $\beta \in [0, 1)$ raised to the power of the position this value has in the descending list. A β value of 0 means that only the most visually polluting UAV has an impact on a ground square within a given time interval. As β goes towards 1, the impact of each additional UAV becomes less and less discounted and thus closer to the full impact of the UAV. We require $\beta < 1$, so that each additional UAV has a diminishing effect on the total visual pollution. This is due to the fact that the human ability to recognize a quantity without counting stops at about 3 to 4 (Hasak & Toomarian, 2022). As an example, the human brain can distinguish between 2 and 5 UAVs, but the difference between 10 and 13 UAVs is more challenging to grasp. Finally, the total visual pollution P_g^V is the sum of the discounted visual pollution values for each time interval. Note that with a higher δ value, the objective value for visual pollution will tend to be lower since more values can be discounted within the same time interval. Also note that a given UAV u may have multiple t_{uis} values in the same time

Algorithm 1 Algorithm to find the total visual pollution generated by a system flight plan in a given ground square g

```
    Input: System flight plan $\vec{t}$, ground square $g$
    Output: Visual pollution in ground square $g$, $P^V_g$
 1: function $P^V(\vec{t}, g)$
 2:     $P^V_g \leftarrow 0$
 3:     $t^{Max} \leftarrow \max\{t_{uis} \in \vec{t}\} - \min\{t_{uis} \in \vec{t}\}$
 4:     $\mathcal{T} \leftarrow \{0, \delta, 2\delta, ..., \lfloor \frac{t^{Max}}{\delta} \rfloor \delta\}$
 5:     for $\tau \in \mathcal{T}$ do
 6:         $P^V_{g\tau} \leftarrow []$
 7:         for $\{t_{uis} \in \vec{t} \mid \exists_{u\in\mathcal{U},\, i\in\mathcal{N},\, s\in\mathcal{S}_u}\ t_{uis} \in [\tau, \tau+\delta) \wedge g \in \mathcal{G}^V_{ui}\ \}$ do
 8:             $P^V_{g\tau}.append\Big((1-F_g)\cdot(C^{V,1}\cdot\frac{1}{D(t_{uis},g)^{C^{V,2}}})\cdot Q_g\Big)$
 9:         end for
10:         $P^V_{g\tau} \leftarrow DescendingSort(P^V_{g\tau})$
11:         for $n \in \{0, 1, ..., |P^V_{g\tau}| - 1\}$ do
12:             $P^V_g \leftarrow P^V_g + P^V_{g\tau}[n]\cdot\beta^n$
13:         end for
14:     end for
15:     return $P^V_g$
16: end function
```

interval. Since traversing a diagonal edge takes more time than traversing a straight edge, more vertices end up in the same time interval if only straight edges are used. Furthermore, two t_{uis} values with an absolute difference smaller than δ could potentially end up in different time intervals. Thus, the use of non-overlapping time intervals is a simplification. It should also be noted that the set of time intervals $\mathcal{T}$ defined in line 4 defines the earliest UAV start time as time 0.

The total amount of visual pollution generated by a system flight plan $\vec{t}$, is thus given as the total visual pollution generated in all ground squares as follows

$$P^V(\vec{t}) = \sum_{g\in\mathcal{G}} P^V(\vec{t}, g) \tag{6.9}$$

6.5 Noise Pollution Model

We have chosen to build a model for noise pollution using sound pressure level (SPL). SPL is a commonly used measure for noise levels with great presence in both literature and empirical data, and has good correlation with human perception of loudness (Long, 2014). SPL is distance dependent (Kjær, n.d.), and hence fits well with the quantification of how noise propagates from UAVs. Empirical research shows that the smallest angle between a UAV, an observing individual and the horizontal has a significant impact on the SPL at the position of the individual (Schäffer, Pieren, Heutschi et al., 2021). The SPL is highest directly below and above the UAV, i.e. at a 90° angle. The change in SPL due to changes in the aforementioned angle varies somewhat differently depending on the frequencies emitted by the UAV, but can in general be said to change linearly with the change in angle (Heutschi et al., 2020; Treichel & Körper, 2019). Furthermore, a review by Schäffer, Pieren, Heutschi et al. (2021) finds that there is a positive but decreasing relationship between UAV weight and the SPL (Schäffer, Pieren, Heutschi et al., 2021). The effect of carrying payloads with different weights is disregarded in this thesis, however.

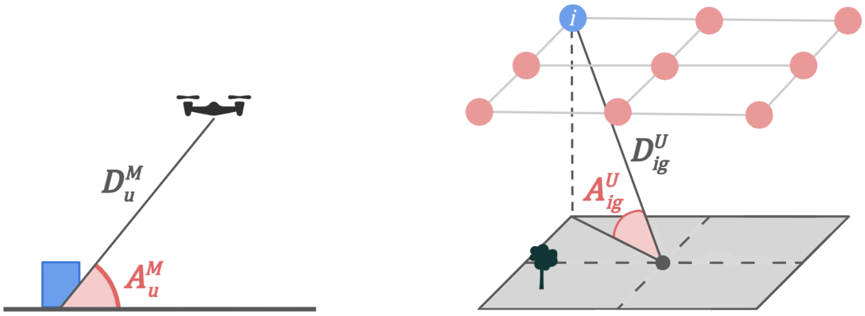


Figure 6.6: Illustration of the angles and distances used to calculate noise pollution. In the left part of the figure, the blue box represents a measurement device. A^M and A_{ig}^U represent angels, and D^M and D_{ig}^U represent distances. The ground square multiple λ is set equal to 2, leading to one ground square below the graph in the right part of the figure. The dashed lines illustrate the projection of the graph onto the ground squares. The diagonal edges have been omitted for readability.

A basic equation for the SPL, measured in decibels, experienced in the center of ground square g when a UAV is positioned in vertex i can be formulated as follows

$$SPL_{ig} = M^S + O \cdot (90 - A^M) + 20 \cdot \log_{10}(D^M) - O \cdot (90 - A_{ig}^U) - 20 \cdot \log_{10}(D_{ig}^U) \tag{6.10}$$

Equation (6.10) uses a measured SPL value from a UAV, M^S, as a starting point. The second and third term convert the value to the corresponding SPL vertically downwards at a distance 1 meter away from the UAV by adjusting for the angle, A^M, and distance, D^M, at which M^S was measured. This is shown in the left part of Figure 6.6. The two last terms further adjust for the angle A_{ig}^U and the distance D_{ig}^U to give the appropriate SPL value at the center of ground square g. This is shown in the right part of Figure 6.6. The angle adjustments are based on the linear relationship between change in SPL value and change in angle. A^M is defined as the smallest angle between the UAV, the horizontal and the measurement device, while A_{ig}^{U} is the smallest angle between vertex i, the horizontal and the center of ground square g. The distance adjustments are based on physical laws for sound propagation in space when considering the sound from a UAV as a point source.

Since decibel is a logarithmic unit, linear addition of decibel values does not make sense. We hence need to make use of the relationship between SPL and the pressure caused by the sound wave, p, which is given as

$$SPL = 20 \cdot \log_{10}\left(\frac{p}{p_0}\right) \Longleftrightarrow p = p_0 \cdot 10^{\frac{SPL}{20}} \tag{6.11}$$

where p_0 is the pressure at the hearing threshold and denotes the lowest sound a human can hear (European Commission, 2008).
Similar to the models for risk and visual pollution, the quantification of noise pollution takes both a single-UAV perspective and a system perspective. Again, we disregard possible subjective differences in perception between individuals.

6.5.1 Single Unmanned Aerial Vehicle Perspective

To quantify the total noise pollution from UAV u when it is positioned in vertex i, we use the following equation

$$\begin{aligned} P_{ui}^{N} &= \sum_{g \in \mathcal{G}_{ui}^{N}} 2^{2\cdot\log_{10}\left(\frac{p}{p_0}\right)} \cdot Q_g = \sum_{g \in \mathcal{G}_{ui}^{N}} 2^{2\cdot\log_{10}\left(\frac{p_0 \cdot 10^{\frac{SPL_{uig}}{20}}}{p_0}\right)} \cdot Q_g = \sum_{g \in \mathcal{G}_{ui}^{N}} 2^{\frac{SPL_{uig}}{10}} \cdot Q_g \\ &= \sum_{g \in \mathcal{G}_{ui}^{N}} 2^{\frac{M_u^S + O\cdot(90 - A_u^M) + 20\cdot\log_{10}(D_u^M) - O\cdot(90 - A_{ig}^U) - 20\cdot\log_{10}(D_{ig}^U)}{10}} \cdot Q_g \end{aligned} \tag{6.12}$$

where $\mathcal{G}_{ui}^{N} \subseteq \mathcal{G}$ is the set of ground squares affected by noise pollution from the UAV. Equation (6.12) combines equation (6.10) and the common statement that a 10 dB sound increase is perceived twice as loud (Salford Acoustics, n.d.), which corresponds to a $\sqrt{10}$ times higher pressure. The logic behind the equation is to find the ratio between the pressure p experienced in a given ground square and the pressure at the hearing threshold, p_0. By raising 2 to the power of 2 multiplied by the base-10 logarithm of this ratio, we find how many times higher than the hearing threshold the sound is perceived. Finally, we multiply the value by the number of individuals in each affected ground square, Q_g, and simplify the math. The noise pollution values P_{ui}^{N} do not have a clear interpretation in themselves, but they are linearly comparable. Thus, they are appropriate for saying how many times worse it is to include one vertex in a UAV's path compared to another.

To find the affected ground squares, we set a cut-off threshold in decibels, R^N, where noise below the threshold level is regarded as no noise pollution. Similar to the model for visual pollution, we create a sphere around the UAV with a radius equal to a cut-off distance, D_u^N, i.e. the maximum distance from the UAV where noise from the UAV *could* be above the threshold R^N. Note that although the noise pollution in a certain ground square depends on the both the distance D_{ig}^U and angle A_{ig}^U relative to the vertex the UAV is in, we find the cut-off distance using only change in distance to ensure that the corresponding sphere covers all potentially affected ground squares. The affected ground squares are then found by checking whether the SPL in the center of each ground square within the sphere is above R^N. This check is necessary since the angle adjustment may lead to the SPL being below R^N although the distance from the UAV is less than D_u^N. The cut-off distance D_u^N is found by solving the following equation

$$\begin{aligned} &M_u^S + O \cdot (90 - A_u^M) + 20 \cdot \log_{10}(D_u^M) - 20 \cdot \log_{10}(D_u^N) = R^N \\ &\quad \Rightarrow D_u^N = 10^{\frac{M_u^S + O \cdot (90 - A_u^M) + 20 \cdot \log_{10}(D_u^M) - R^N}{20}} \end{aligned} \tag{6.13}$$

Similar to the model of visual pollution, we use the noise pollution value of the vertex a UAV leaves from, and include the value associated with the final vertex on each segment, when finding the total noise pollution. Thus, the total noise pollution of a given path for a single UAV u is given as

$$P_u^N = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} P_{ui}^N \cdot x_{uijs} + \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_u} P_{ui}^N \cdot I_{uis}^F \tag{6.14}$$

6.5.2 System Perspective

To quantify the total noise pollution generated by a system flight plan, we first find the noise pollution in each ground square g using Algorithm 2. The appropriate decibel value of the simultaneous impact from a system of UAVs can be found by converting the decibel values to pressure, summing the square of the pressures, and converting the square root of this sum back to decibels (Baker, n.d.). This implies that the sum of the noise from experiencing multiple UAVs separately is greater than the noise from experiencing them simultaneously.

Algorithm 2 divides the entire time frame of system flight plan $\vec{t}$ into a set of time intervals of a given length δ. The earliest start time among the UAVs in the system is set as time 0 in line 4. The for-loop in lines 7-9 sums up the squared pressures from the UAVs in a given time interval, using the conversion from SPL to pressure shown in equation (6.11). The expression in the exponent in line 8 is similar to equation (6.10), except for the added u index, and that A_{ig}^u and D_{ig}^u are replaced with the functions $A(t_{uis}, g)$ and $D(t_{uis}, g)$, respectively, which return the same values. In line 10, the resulting SPL from the system of UAVs is found by converting back to decibels according to equation (6.11). The if-statement in line 11 checks whether the resulting decibel value is above the threshold. If so, the appropriate noise pollution value for the time interval is found using equation (6.12) for ground square g and linearly added to the total noise pollution value, P_g^N, in line 12. Note that a change in the value of δ can affect

Algorithm 2 Algorithm to find the total noise pollution generated by a system flight plan in a given ground square g

```
    **Input**: System flight plan $\vec{t}$, Ground square $g$
    **Output**: Noise pollution in ground square $g$, $P_g^N$
 1: **function** $P^N(\vec{t},\ g)$
 2:     $P_g^N \leftarrow 0$
 3:     $t^{Max} \leftarrow \max\{t_{uis} \in \vec{t}\} - \min\{t_{uis} \in \vec{t}\}$
 4:     $\mathcal{T} \leftarrow \{0, \delta, 2\delta, ..., \lfloor \frac{t^{Max}}{\delta} \rfloor \delta\}$
 5:     **for** $\tau \in \mathcal{T}$ **do**
 6:         $P_{gt}^N \leftarrow 0$
 7:         **for** $\{t_{uis} \in \vec{t} \mid \exists_{u \in \mathcal{U},\, i \in \mathcal{N},\, s \in \mathcal{S}_u}\ t_{uis} \in [\tau, \tau + \delta) \wedge g \in \mathcal{G}_{ui}^{NS}\ \}$ **do**
 8:             $P_{g\tau}^N \leftarrow P_{g\tau}^N + \left(p_0 \cdot 10^{\frac{M_u^S + O \cdot (90 - A_u^M) + 20 \cdot \log_{10}(D_u^M) - O \cdot (90 - A(t_{uis}, g)) - 20 \cdot \log_{10}(D(t_{uis}, g))}{20}}\right)^2$
 9:         **end for**
10:         $SPL_g \leftarrow 20 \cdot \log_{10}\left(\frac{\sqrt{P_{g\tau}^N}}{p_0}\right)$
11:         **if** $SPL_g > R^N$ **then**
12:             $P_g^N \leftarrow P_g^N + 2^{\frac{SPL_g}{10}} \cdot Q_g$
13:         **end if**
14:     **end for**
15:     **return** $P_g^N$
16: **end function**
```

the noise pollution objective value in both directions. If UAVs individually generate noise that exceeds the noise pollution threshold in the center of a ground square, it is beneficial with a higher δ value since more values can be discounted. If, on the other hand, the UAVs do not generate noise exceeding the threshold on their own, but the simultaneous impact from a system does, a higher δ value can increase the objective value. Similar to the model of visual pollution, a UAV may have multiple t_{uis} values in the same time interval and two t_{uis} values with an absolute difference smaller than δ may end up in different time intervals.

A thing to note about Algorithm 2 is that the set $\mathcal{G}_{ui}^{NS} \subseteq \mathcal{G}$ in line 7 is different from the set $\mathcal{G}_{ui}^{N} \subseteq \mathcal{G}$ in equation (6.12). The rationale for this is that a UAV might contribute to the SPL from a system of UAVs exceeding the threshold, although the SPL from the UAV in isolation is below the threshold. Hence, the set of affected ground squares when considering the UAV as part of a system of UAVs, $\mathcal{G}_{ui}^{NS}$, is based on a reduced threshold value compared to R^N, namely $R^N - R^{N,Red}$, and $\mathcal{G}_{ui}^{NS}$ can thus include more ground squares than $\mathcal{G}_{ui}^{N}$. $R^{N,Red}$ is set low enough to ensure that all UAVs that can contribute to the total noise from a system of UAVs exceeding R^N in any ground square are accounted for.

The total amount of noise pollution generated by a system flight plan $\vec{\boldsymbol{t}}$, is thus given as the total noise pollution generated in all ground squares for all time intervals as follows

$$P^N(\vec{\boldsymbol{t}}) = \sum_{g \in \mathcal{G}} P^N(\vec{\boldsymbol{t}}, g) \tag{6.15}$$

Chapter 7

Solution Method

This chapter presents the solution method used to solve the static, multi-UAV path planning (MUAVPP) problem, which is a hybrid evolutionary algorithm (HEA). This heuristic approach combines the non-dominated sorting genetic algorithm III (NSGA-III) with ant colony optimization (ACO), and is described in Section 7.1. To better understand runtime and solution quality aspects of the heuristic solution method, this thesis compares the HEA to an exact problem-specific algorithm called multi-objective Dijkstra's algorithm (MDA). The MDA uses dynamic programming to generate the complete Pareto front, and is presented in Section 7.2. The MDA used in this thesis is based on work done in our preparatory research project, but is sped up using a technique proposed by Maristany de las Casas et al. (2021).

7.1 Hybrid Evolutionary Algorithm

This section describes how the static MUAVPP problem described in Chapter 4 and modeled in Chapter 5, is solved using a hybrid evolutionary algorithm. There are several reasons behind our choice of solution method. As described in the literature review in Chapter 3, evolutionary algorithms (EAs) are a common solution method, especially for more complex problems with multiple objectives or a three dimensional environment. Since the MUAVPP problem considers both of these elements, in addition to e.g. multiple vehicles, the use of an EA seems reasonable. Furthermore, since generating good system flight plans is adequate and optimality is not a necessity, the choice of an EA as a solution method is strengthened. An EA can also be more efficient than an exact method, because it does not have to find the entire Pareto front. Moreover, a decision maker will have no use for hundreds of choices, and generating the entire Pareto front with an exact method is thus not of practical value. It should also be noted that the system perspective equations for the risk, visual pollution and noise pollution objectives are more complex than the single-UAV perspective equations described in Chapter 6. Hence, an efficient solution method is essential. Finally, an EA may be stopped at any point in time and return viable solutions, which again supports the efficiency argument.

The HEA used in this thesis consists of NSGA-III with ACO as a subroutine. NSGA-III and ACO are described in Subsections 7.1.1 and 7.1.2, respectively.

7.1.1 Non-Dominated Sorting Genetic Algorithm III

The NSGA-III used as the basis for the HEA, is heavily inspired by the original NSGA-III developed by Deb and Jain (2014). There are multiple aspects of NSGA-III that make it suitable for solving the MUAVPP problem. First, genetic algorithms are a common type of EA among the analyzed papers in the literature review in Chapter 3. Furthermore, genetic algorithms produce a *set* of solutions in a single run, since such algorithms evolve an entire population of solutions in each iteration. Since the MUAVPP problem is a multi-objective problem, generating a few system flight plans with various trade-offs between the objectives is essential as it gives the decision maker a realistically sized set of options to choose from. The choice of a *non-dominated sorting* genetic algorithm in particular is also motivated by the problem studied in this thesis having multiple objectives, making Pareto dominance tests essential to decide which system flight plans are better than others. The use of the NSGA-III version of NSGA, is due to NSGA-III being more well-suited, and thus more efficient, as the number of objectives in a multi-objective problem increases compared to NSGA and NSGA-II. This is due to NSGA-III using *reference points* to select individuals for the next iteration (Deb & Jain, 2014).

The implementation of the NSGA-III-based HEA is shown in Algorithm 3 and visualized as a flowchart in Figure 7.1. The algorithm begins by creating an initial parent population $\mathcal{P}$ containing N individuals. Each *individual* in the population is a system flight plan, i.e. one path with associated timestamps for each UAV in the system. The $isSystem$ variable denotes whether the problem to be solved considers a system of UAVs and is used to determine how the initial parent population is created in lines 2-10. When $isSystem$ is $True$, the algorithm recursively calls itself once for each UAV in the system, with $isSystem = False$ and $M = N = \lceil \frac{N}{|\mathcal{U}|} \rceil$ in lines 3-5. These $|\mathcal{U}|$ initializations of *non-system*, i.e. single-UAV, HEA runs is the reason why there is a *set* of arrows between the "Create initial population" and "Initialize non-system HEA" boxes at the top of the flowchart in Figure 7.1. The non-system runs of the HEA lead to a set of flight plans for each UAV individually, and these results are merged in line 6 in Algorithm 3. Random combinations of one path for each UAV are used to create system flight plans for the initial population in the $createInitialSystemPopulation$ method in line 7. This is visualized in the large blue box at the top-right of the flowchart in Figure 7.1. When $isSystem$ is $False$, the algorithm uses ACO to create the initial population in line 9. The intermediate non-system

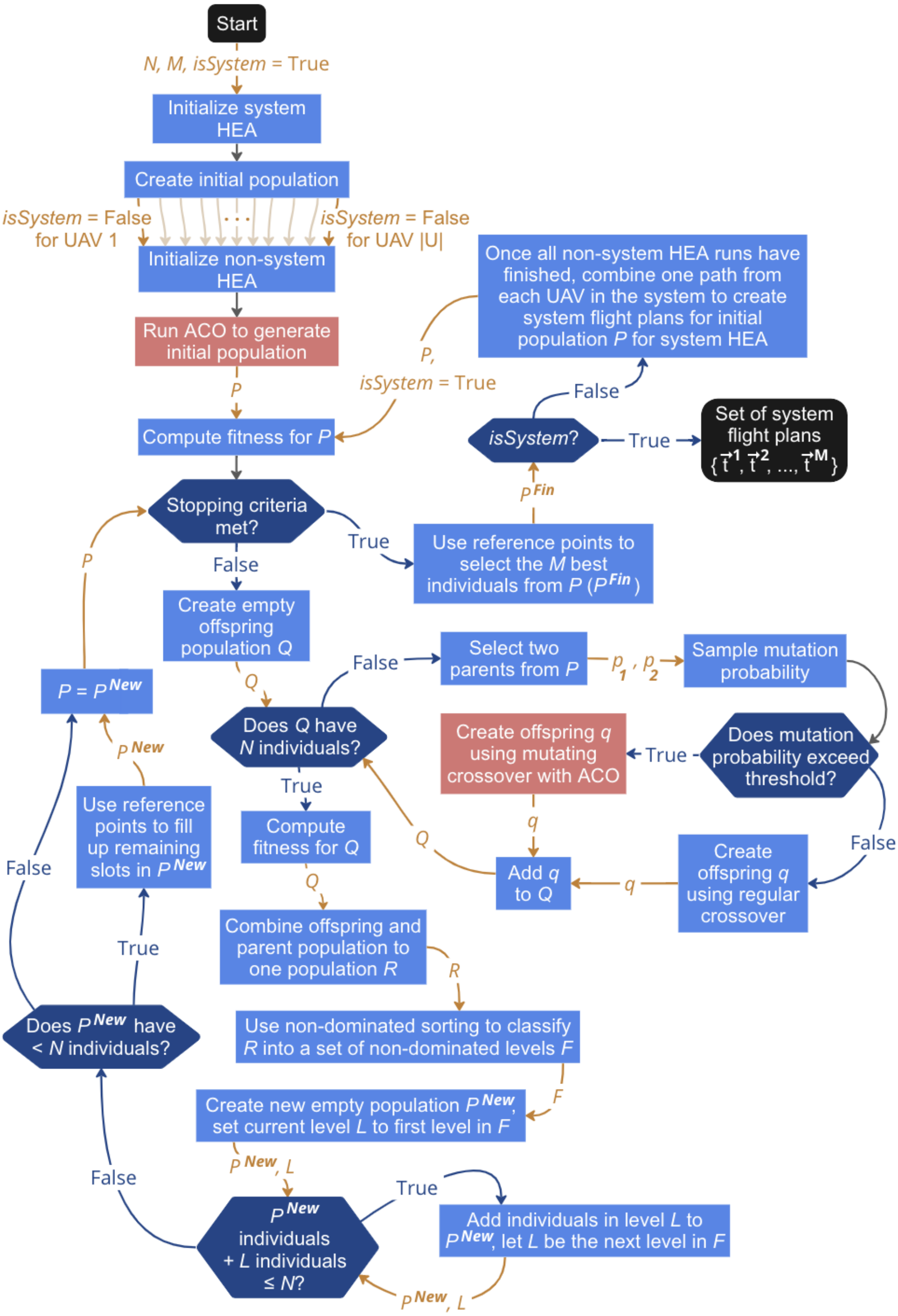

Start
N, M, isSystem = True
Initialize system HEA
Create initial population
isSystem = False for UAV 1
. . .
isSystem = False for UAV |U|
Initialize non-system HEA
Run ACO to generate initial population
P
Compute fitness for P
Once all non-system HEA runs have finished, combine one path from each UAV in the system to create system flight plans for initial population P for system HEA
P, isSystem = True
False
isSystem?
True
Set of system flight plans $\{\vec{\tau}^1, \vec{\tau}^2, ..., \vec{\tau}^M\}$
P^{Fin}
Stopping criteria met?
True
Use reference points to select the M best individuals from P (P^{Fin})
P
False
Create empty offspring population Q
$P = P^{New}$
Q
False
Select two parents from P
p_1, p_2
Sample mutation probability
Does Q have N individuals?
P^{New}
Create offspring q using mutating crossover with ACO
True
Does mutation probability exceed threshold?
True
Use reference points to fill up remaining slots in P^{New}
Compute fitness for Q
Q
q
False
False
Add q to Q
q
Create offspring q using regular crossover
Q
True
Combine offspring and parent population to one population R
R
Does P^{New} have < N individuals?
Use non-dominated sorting to classify R into a set of non-dominated levels F
F
Create new empty population P^{New}, set current level L to first level in F
False
P^{New}, L
True
P^{New} individuals + L individuals ≤ N?
Add individuals in level L to P^{New}, let L be the next level in F
P^{New}, L

Figure 7.1: A flowchart for the hybrid evolutionary algorithm (HEA) for a system of UAVs. Blue boxes are steps in the NSGA-III base, while dark blue hexagons are if-statements with corresponding dark blue result arrows. The light red boxes are steps using ACO. The two black, rounded boxes correspond to the start and end points of the algorithm. Orange arrows means the value on the arrow was created or updated by the box the arrow originates from.

HEA runs are included to improve the paths generated by the ACO runs before they are used as the initial population in the system HEA run. After generating the initial parent population $\mathcal{P}$, Algorithm 3 computes the fitness for each individual in $\mathcal{P}$. The $computeFitness$ method uses the single-UAV perspective equations in Chapter 6 if $isSystem$ is $False$, and the system perspective equations if $isSystem$ is $True$. The method associates a fitness vector [*Flight time, Risk, Visual pollution, Noise pollution*], i.e. a vector with one value for each objective, with each individual in $\mathcal{P}$.

The main part of Algorithm 3 is the while loop in lines 12-27. In short, each iteration generates an offspring population $\mathcal{Q}$ of size N. Then the N most fit individuals from the combined parent and offspring population of size $2N$ are selected to be in the parent population $\mathcal{P}$ in the next iteration of the algorithm. The while loop continues until a given stopping criteria is met. The stopping criteria used in the HEA is based on the percentage improvement of the *average fitness vector* of a population compared to its predecessor population, and is given as

$$\exists_{c \in \mathcal{C}}\, c \geq c^{Min}, \qquad \mathcal{C} = \left\{ \frac{\overline{T}^{prev} - \overline{T}}{\overline{T}^{prev}}, \frac{\overline{R}^{prev} - \overline{R}}{\overline{R}^{prev}}, \frac{\overline{P}^{V,prev} - \overline{P}^{V}}{\overline{P}^{V,prev}}, \frac{\overline{P}^{N,prev} - \overline{P}^{N}}{\overline{P}^{N,prev}} \right\} \tag{7.1}$$

Equation (7.1) states that if the population's average values for one or more objectives are at least a percentage threshold c^{Min} lower than the corresponding objective values in the previous iteration, the while loop continues for another iteration. A minimum number of iterations to run before the improvement percentage becomes a requirement can also be specified.

The $createOffspring$ method in line 13 in Algorithm 3 is used to create an offspring population $\mathcal{Q}$ of size N in each iteration. This method is explained in detail at the end of this subsection. Once an offspring population has been created, the algorithm computes the fitness vectors of the offspring population in line 14, before combining the parent and offspring populations in line 15. Then the $nonDominatedSort$ method is used to perform non-dominated sorting on the combined population $\mathcal{R}$. The implementation of $nonDominatedSort$ exactly follows the common implementation found in the original NSGA (Srinivas & Deb, 1994), NSGA-II (Deb, Agrawal et al., 2000) and NSGA-III (Deb & Jain, 2014) papers. For each individual in $\mathcal{R}$, the method checks which individuals dominate, and are dominated by, the given individual based on the fitness vectors. This information is then used to sort the individuals into *non-dominated levels*, where the individuals in each level are only dominated by individuals in a higher level. This means that level 1 individuals are not dominated by any other individuals in $\mathcal{R}$, level 2 individuals are only dominated by individuals in level 1, level 3 individuals are dominated by individuals in level 1 and 2, and so on. After the non-dominated sorting, a new empty population is created, P^{New}. Then the while loop in lines 19-22 in Algorithm 3 adds entire levels from the combined and non-dominated-sorted population $\mathcal{F}$ to P^{New}. It begins by adding level 1, and continues until the addition of another level $\mathcal{L}$ would lead to the number of individuals

Algorithm 3 NSGA-III-based hybrid evolutionary algorithm to generate a set of system flight plans

Input: Set of UAVs $\mathcal{U}$, population size N, number of system flight plans to generate M ($M \leq N$), whether the problem considers a system of UAVs $isSystem$

Output: Set of system flight plans, $\{\vec{t}^{\,1}, \vec{t}^{\,2}, ..., \vec{t}^{\,M}\}$

```
 1: function HEA($\mathcal{U}$, $N$, $M$, $isSystem$)
 2:     if $isSystem$ then
 3:         for $u \in \mathcal{U}$ do
 4:             $\mathcal{P}_u \leftarrow$ HEA($\{u\}$, $\lceil \frac{N}{|\mathcal{U}|} \rceil$, $\lceil \frac{N}{|\mathcal{U}|} \rceil$, $False$)
 5:         end for
 6:         $\mathcal{P}^{All} \leftarrow \bigcup_{u \in \mathcal{U}} \mathcal{P}_u$
 7:         $\mathcal{P} \leftarrow createInitialSystemPopulation(\mathcal{P}^{All}, N)$
 8:     else
 9:         $\mathcal{P} \leftarrow createInitialPopulationWithACO(\mathcal{U}, N)$
10:     end if
11:     $computeFitness(\mathcal{P}, isSystem)$
12:     while stopping criteria not met do
13:         $\mathcal{Q} \leftarrow createOffspring(\mathcal{U}, \mathcal{P}, N, isSystem)$
14:         $computeFitness(\mathcal{Q}, isSystem)$
15:         $\mathcal{R} \leftarrow \mathcal{P} \cup \mathcal{Q}$
16:         $\mathcal{F} \leftarrow nonDominatedSort(\mathcal{R})$
17:         $\mathcal{P}^{New} \leftarrow \emptyset$
18:         $\mathcal{L} \leftarrow getNextLevel(\mathcal{F})$
19:         while $|\mathcal{P}^{New}| + |\mathcal{L}| \leq N$ do
20:             $\mathcal{P}^{New} \leftarrow \mathcal{P}^{New} \cup \mathcal{L}$
21:             $\mathcal{L} \leftarrow getNextLevel(\mathcal{F})$
22:         end while
23:         if $|\mathcal{P}^{New}| < N$ then
24:             $\mathcal{P}^{New} \leftarrow \mathcal{P}^{New} \cup referencePointBasedAddition(\mathcal{P}^{New}, \mathcal{L}, N - |\mathcal{P}^{New}|)$
25:         end if
26:         $\mathcal{P} \leftarrow \mathcal{P}^{New}$
27:     end while
28:     $\mathcal{P}^{Fin} \leftarrow finalSelectionReferencePointBasedAddition(\mathcal{P}, M)$
29:     return $\mathcal{P}^{Fin}$
30: end function
```

in P^{New} becoming strictly larger than N. The algorithm then checks if there are less than N individuals in P^{New} in line 23, and uses the $referencePointBasedAddition$ method to decide which individuals from the level $\mathcal{L}$ to add to P^{New} to reach a population size of N if needed.

The implementation of the $referencePointBasedAddition$ method follows NSGA-III (Deb & Jain, 2014) exactly. The method begins by generating evenly distributed reference points on a normalized hyper-plane that intercepts each of the objective axes in 1 and are equally inclined to all axes. An example is given in Figure 7.2. The number of reference points is given by

$$No.\ of\ reference\ points = \binom{No.\ of\ objectives + d - 1}{d} \tag{7.2}$$

where d is the number of divisions made along each objective axis. Each reference point consists of a set of four values, one for each of the objectives. Each value is a multiple of $1/d$, and the sum of the values in a reference point is 1. The observant reader may notice that this implies that the set of reference points may be viewed as a *set of weightings* for the four objectives, with each of the four values in a reference point corresponding to the weight given to each of the objectives.

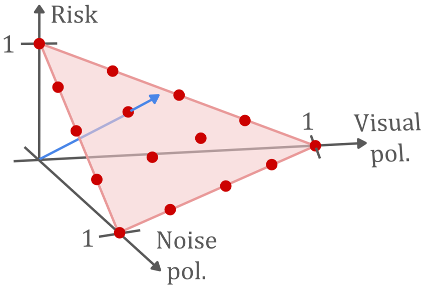


Figure 7.2: Illustration of reference points for a tri-objective problem with the number of divisions $d = 4$. The red dots are reference points, the light red triangle is the normalized hyperplane, and the blue line is the reference line for the reference point it goes through.

After creating the reference points, the $referencePointBasedAddition$ method normalizes the fitness vectors of each individual so all the objective values are between 0 and 1. Each of the normalized fitness vectors are then *associated* with a reference point. This is done by defining *reference lines* between each reference point and the origin, and assigning each normalized fitness vector to the reference point corresponding to the reference line the vector is closest to in the normalized objective space. Each reference point is then given a *niche count*, which is the number of individuals in $\mathcal{P}^{New}$ that is associated with the given reference point. Finally, $\mathcal{P}^{New}$ is increased to N individuals by repeatedly finding the reference point with the lowest niche count, with ties being broken by random choice, and checking which individuals from level $\mathcal{L}$ are associated with this reference point. If multiple such individuals exist, the individual with the minimum distance to the reference line corresponding to the chosen reference point is added to $\mathcal{P}^{New}$. Then the niche count of the chosen reference point is increased by 1. If no individual from level $\mathcal{L}$ is associated with the chosen reference point, the reference point is discarded for the rest of the run of the $referencePointBasedAddition$ method. Once $\mathcal{P}^{New}$ reaches a size of N individuals, the method returns the updated $\mathcal{P}^{New}$. An important strength of using reference points is that it attempts to maintain diversity among the individuals, by adding individuals that are associated with reference points with the lowest niche counts first.

After filling up any remaining slots to ensure the new parent population $\mathcal{P}^{New}$ has N individuals, $\mathcal{P}^{New}$ becomes the parent population $\mathcal{P}$ in the next iteration of the while loop in lines 12-27. Once the stopping criteria is met, the $finalSelectionReferencePointBasedAddition$ method reduces $\mathcal{P}$ down to M individuals if $M < N$. The method uses $referencePointBasedAddition$ to select individuals from level 1, i.e. non-dominated individuals, if there are M or more level 1 individuals. Otherwise it supplements with lower level individuals to reach M individuals. If $M = N$, the method just returns $\mathcal{P}$. The use of reference points in the final selection is done to ensure diversity in the final set of individuals $\mathcal{P}^{Fin}$ returned from Algorithm 3. Note that if $M < N$, the number of reference points is reduced to fit the reduced number of individuals to select in the $finalSelectionReferencePointBasedAddition$ method compared to the number of reference points used in the $referencePointBasedAddition$ method in line 24.

Creation of Offspring

The $createOffspring$ method used to generate an offspring population $\mathcal{Q}$ based on a parent population $\mathcal{P}$, is shown in Algorithm 4. Until N energy feasible offspring have been created, the while loop in lines 3-17 selects two individuals from the parent population $\mathcal{P}$ using the $selectParents$ method. For each of the two parents to select, the $selectParents$ method selects a parent at random from *all* individuals in $\mathcal{P}$ with a given probability, otherwise a random choice of parent is made among the *non-dominated* individuals in $\mathcal{P}$. After selecting parent individuals p_1 and p_2, one of the UAVs in the system is chosen at random to have its path updated in line 5. The reason behind only updating the path of one of the UAVs when creating an offspring, is that updating the paths of multiple UAVs at once could lead to a fitness improving change in one UAV's path being offset by a worsening change in another UAV's path. Having chosen the UAV to update the path of, a new offspring is created by using $mutatingCrossover$ with a given probability, otherwise $regularCrossover$ is used. If $isSystem = True$, any collisions are fixed after the crossover.

The $mutatingCrossover$ method is shown in the upper right part of Figure 7.3. It begins by randomly selecting *one* segment, s, on the path of UAV u. For all segments before segment s, the segment path of parent p_1 is used, while for all segments after segment s, the segment path of parent p_2 is used. For the chosen segment s however, a random portion of the *start* of the segment path of p_1 is used and a random portion of the *end* of the segment path of p_2 is used, leaving a gap between the two. Then ACO is used to connect the two partial paths on segment s. Since the point in time when UAV u reaches the various locations in p_1 and p_2 may vary, all the t_{uis} values from the vertex where the ACO path begins and throughout the rest of the total path must be updated. After the timing update, the $mutatingCrossover$

Algorithm 4 Algorithm to create offspring for an iteration of the HEA

```
    Input: Set of UAVs $\mathcal{U}$, parent population $\mathcal{P}$, population size $N$, whether the problem
        considers a system of UAVs $isSystem$
    Output: An offspring population, $\mathcal{Q}$
 1: function CreateOffspring($\mathcal{U}$, $\mathcal{P}$, $N$, $isSystem$)
 2:     $\mathcal{Q} \leftarrow \emptyset$
 3:     while $|\mathcal{Q}| < N$ do
 4:         $p_1, p_2 \leftarrow selectParents(\mathcal{P})$
 5:         $u \leftarrow randomChoice(\mathcal{U})$
 6:         if $randomFloat(0,1) \geq mutationThreshold$ or $|\mathcal{S}_u| = 1$ then
 7:             $offspring \leftarrow mutatingCrossover(p_1, p_2, u)$
 8:         else
 9:             $offspring \leftarrow regularCrossover(p_1, p_2, u)$
10:         end if
11:         if $checkIsEnergyFeasible(offspring)$ then
12:             if $isSystem$ then
13:                 $offspring \leftarrow fixPotentialCollisions(offspring)$
14:             end if
15:             $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{offspring\}$
16:         end if
17:     end while
18:     return $\mathcal{Q}$
19: end function
```

method combines the new path for UAV u with paths for the remaining UAVs to create a new individual. All the remaining UAVs' paths are taken from one of the parents, with the parent being chosen at random. The new individual is then returned.

The $regularCrossover$ method works by going through *each* segment in the path of UAV u and randomly choosing between the segment path of parent p_1 or p_2, ensuring that each parent is chosen at least once. Since the method combines segment paths from the two parents, a UAV with only one segment are not able to use the $regularCrossover$ method, which is why the size of the segment set of the chosen UAV u, $|\mathcal{S}_u|$, is checked in line 6 of Algorithm 4. Similar to the $mutatingCrossover$ method, updates must be made to a subset of the t_{uis} values. All t_{uis} values from the start of the second segment, or a later segment if the same parent is used for consecutive segments in the beginning, and throughout the rest of the total path must be updated. Then the paths from p_1 are used for the remaining UAVs to create a new individual that is subsequently returned. The selection of which parent the remaining paths are taken from is essentially random, since p_1 is randomly selected from either the entire parent population $\mathcal{P}$ or from the non-dominated subset of $\mathcal{P}$. The $regularCrossover$ method is illustrated in the lower right part of Figure 7.3.

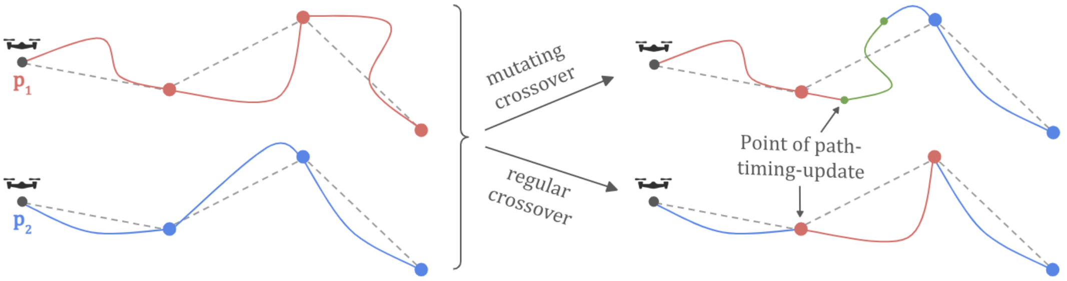


Figure 7.3: Illustration of the two crossover types used in the hybrid evolutionary algorithm. The left part of the figure shows the paths for UAV u for parents p_1 and p_2, with the red and blue dots corresponding to locations to visit. The upper right part shows a mutation crossover on the second segment. The red parts of the path are from p_1 while the blue parts are from p_2. The green part of the path is the mutated path generated by ACO, and the small green dots are the start and end vertices of the ACO path. The lower right part shows a regular crossover, where the first and third segment are taken from p_2 while the second segment is taken from p_1. The first green dot in the mutation crossover path and the first red dot in the regular crossover path correspond to the starting points for where timing updates must be made to the rest of the paths.

After a new offspring has been created, its energy feasibility is checked in line 11 of Algorithm 4. If the offspring is energy feasible, i.e. constraints (5.12) in the MINLP model described in Chapter 5 hold, the offspring is added to the offspring population $\mathcal{Q}$. If the problem is a multi-UAV problem, any collisions occurring due to the selected UAV's updated path are fixed in line 13 using the $fixPotentialCollisions$ method, before the offspring is added to $\mathcal{Q}$. The $fixPotentialCollisions$ method greedily finds the shortest collision-free path from the final vertex before a collision to the first vertex after the collision, with "collision vertices" defined as any vertex in a UAV's path that violates the collision constraints (5.9)-(5.11) in the MINLP model. Thus, it is the UAV selected to be updated that is responsible for avoiding a collision due to its new path. Then the while loop in lines 3-17 begins another iteration.

7.1.2 Ant Colony Optimization Subroutine

The HEA uses ACO to generate paths for a single UAV, either as part of the generation of an initial parent population or as part of the mutating crossover, as described in Subsection 7.1.1. The implementation of ACO takes inspiration from several articles, but is mainly based on the original ACO algorithm proposed by Colorni et al. (1991). The main motivation behind the use of ACO as a subroutine, is that it is a simple and efficient way to generate UAV paths.

The use of a genetic algorithm requires an initial population of system flight plans. Using ACO allows for a more guided search for good UAV paths than for example random path generation. In addition, the use of ACO in mutating crossovers introduces completely new paths into the system flight plans, which helps the HEA to break out of local optima. Furthermore, ACO was found to be a common EA for path planning and similar problems in the literature review in Chapter 3, although not as common as genetic algorithms.

Algorithm 5 Ant colony optimization algorithm for finding paths for a single UAV

```
Input: Maximum number of iterations $I$, total number of ants $A$, set of start vertices $\mathcal{V}^S$,
       set of end vertices $\mathcal{V}^E$, number of paths to create $P$
Output: A set of $P$ paths for a single UAV
function ACO($I, A, \mathcal{V}^S, \mathcal{V}^E, P$)
    $bestPaths \leftarrow \emptyset$
    $initializePheromones(\mathcal{V}^S)$
    for $\zeta \in \{1, 2, ..., I\}$ do
        $antPaths \leftarrow [[v^S_1], [v^S_2], ..., [v^S_A]]$
        while unfinished ants exist do
            for $a \in unfinishedAnts$ do
                $v^{Next} \leftarrow selectNextVertex(antPaths[a])$
                if $v^{Next}$ is None then
                    $antPaths[a] \leftarrow [v^S_a]$
                else
                    $antPaths[a].insert(v^{Next})$
                end if
                $onlinePheromoneUpdate(antPaths[a], v^{Next})$
            end for
            $unfinishedAnts \leftarrow getUnfinishedAntPaths(antPaths, \mathcal{V}^E)$
        end while
        $evaluateSolutions(antPaths, \mathcal{V}^S)$
        $offlinePheromoneUpdate(antPaths, \mathcal{V}^S)$
        $possibleBestPaths \leftarrow getEnergyFeasiblePaths(antPaths, \mathcal{V}^S)$
        $bestPaths \leftarrow updateBestPaths(bestPaths, possibleBestPaths, P)$
    end for
    if not $v^S_1 = v^S_2 = ... = v^S_A$ then
        $bestPaths \leftarrow combineSegmentPaths(bestPaths)$
    end if
    return $bestPaths$
end function
```

The ACO implementation used in this thesis is shown in Algorithm 5. In short, the algorithm creates A ant paths in each iteration and updates the set of best paths if better paths are found in the given iteration. After I iterations, it returns the P best paths found. An important thing to note is that in both applications of ACO in the HEA, i.e. in mutating crossover and in the generation of an initial parent population, a given ant will only be used to find a path on *one* segment. This means that when ACO is used for mutation, all A ants traverse the same segment and have the same start and end vertices, i.e. all values in $\mathcal{V}^S$ are equal and all values in $\mathcal{V}^E$ are equal. When ACO is used to create an initial parent population on the other hand, the A ants are spread across the set of segments for the UAV and the values in $\mathcal{V}^S$ and $\mathcal{V}^E$ vary based on the segment a given ant is set to traverse.

Algorithm 5 begins by creating an empty set, $bestPaths$, and initializing the pheromone values of all edges in the graph to a common value, μ_0. It then starts the search for UAV paths in the for-loop in lines 4-22. In each iteration, line 5 places each ant a at its start vertex v_a^S, i.e. creates a path list of only one vertex for each ant. The while-loop in lines 6-17 then runs until every ant has found a path from its start vertex to its end vertex. The for-loop in lines 7-15 goes through each ant a that has yet to reach its end vertex and finds the next vertex in its path using the $selectNextVertex$ method. The $selectNextVertex$ method is based on Colorni et al. (1991). It works by randomly selecting a vertex among the neighboring vertices of the vertex the ant is currently in. The probability of selecting a neighboring vertex j when ant a is positioned in vertex i on segment s, π_{aijs}, is given by

$$\pi_{aijs} = \begin{cases} \frac{(\mu_{ijs})^{\varepsilon} \cdot (\eta_{aj})^{\omega}}{\sum_{l \in \mathcal{N}_i} (\mu_{ils})^{\varepsilon} \cdot (\eta_{al})^{\omega}}, & \text{if ant } a \text{ traverses segment } s \text{ and} j \in \mathcal{N}_i \text{ and } j \notin \mathcal{N}_a^V \\ 0 \quad , & \text{otherwise} \end{cases},$$

$$\text{where } \eta_{aj} = \frac{1}{d(j, v_a^E)} \tag{7.3}$$

where μ_{ijs} is the current pheromone value on edge (i,j) on segment s, η_{aj} is the inverted distance between vertex j and the end vertex v_a^E for ant a, and ε and ω are constants. The segment index on the pheromone values is due to the possibility of not all ants traversing the same segment. Only ants on the same segment should use the same pheromone values, because some of the parameters that affect the four objectives vary based on segment, and thus the same edge can be "good" for one segment and "bad" for another. $\mathcal{N}_a^V$ is the set of vertices already visited by ant a, and is included to avoid the ants going in cycles. The $selectNextVertex$ method will return *None* if all neighboring vertices to vertex i have already been visited. If this occurs, Algorithm 5 restarts the ant at its start vertex in line 10. Otherwise, vertex j, called v^{Next} in the algorithm, is added to the path of ant a in line 12. Then an online pheromone update for edge (i,j) is performed in line 14. This local update is done to increase search diversity, as it reduces the chance of multiple ants taking the same path in a given iteration. The implementation of the $onlinePheromoneUpdate$ method follows Dorigo and Gambardella (1997) exactly, using the following equation to update the pheromone value of edge (i, j) on segment s

$$\mu_{ijs} = (1 - \varphi) \cdot \mu_{ijs} + \varphi \cdot \mu_0 \tag{7.4}$$

where $\varphi \in (0, 1]$ is the pheromone decay rate and μ_0 is the initial pheromone value. After selecting the next vertex for each ant in $unfinishedAnts$, the $unfinishedAnts$ set is updated in line 16 by removing all ants that have reached their end vertices.

Once all A ants have reached their end vertices, the ant paths are evaluated by the $evaluateSolutions$ method. Since the static MUAVPP problem is a multi-objective problem and the original ACO algorithm proposed by Colorni et al. (1991) was created for single-objective problems, the method is implemented based on a multi-objective version of ACO proposed by Ntakolia and Lyridis (2022). The $evaluateSolutions$ method begins by computing the objective values for each ant path using the single-UAV perspective equations from Chapter 6. Then the objectives are normalized and the normalized root mean square error (NRMSE) for each ant paths' objective values is calculated using the following equation

$$NRMSE_a = \frac{1}{4} \cdot \sqrt{(T^{Norm})^2 + (R^{Norm})^2 + (P^{V,Norm})^2 + (P^{N,Norm})^2} \tag{7.5}$$

The resulting NRMSE values are associated with each ant path. If ACO is used to create an initial parent population, the NRMSE values are calculated per segment, i.e. using the maximum and minimum objective values for ants on the same segment as ant a to normalize the objectives. The use of NRMSE for solution evaluation is due to its high convergence speed (Ntakolia & Lyridis, 2022).

Next, an offline pheromone update is performed on all edges in the graph. The $offlinePheromoneUpdate$ method is implemented based on the pheromone differentiated update strategy proposed by Xie et al. (2022). Compared to the offline pheromone update used in the the original ACO algorithm (Colorni et al., 1991), the pheromone differentiated update strategy proposed by Xie et al. (2022) is able to reflect low-NRMSE paths in the pheromone values more quickly and thus increase the efficiency of the search. First, the amount of pheromone left by an ant a on edge (i, j) given that ant a traversed segment s, $\Delta\mu^{a}_{ijs}$, is calculated as follows

$$\Delta\mu_{aijs} = \begin{cases} \frac{NRMSE_s^{Avg} - NRMSE_a}{NRMSE_s^{Avg} - NRMSE_s^{Min}} \cdot \frac{1}{NRMSE_a}, & \text{if } NRMSE_a \leq NRMSE_s^{Avg} \\ -\frac{NRMSE_a - NRMSE_s^{Avg}}{NRMSE_s^{Avg} - NRMSE_s^{Min}} \cdot \frac{1}{NRMSE_s^{Max}}, & \text{if } NRMSE_a > NRMSE_s^{Avg} \end{cases} \tag{7.6}$$

where $NRMSE_s^{Max}$, $NRMSE_s^{Min}$ and $NRMSE_s^{Avg}$ are the maximum, minimum and average NRMSE values of all ants traversing the same segment s as ant a. $\Delta\mu_{aijs}$ is assumed to be zero for all edges and segments ant a did not traverse in the given iteration. Then the segment pheromone value of each edge (i, j) in the graph for each segment s is updated as follows

$$\mu_{ijs} = (1-\rho)\cdot\mu_{ijs} + \rho\cdot\sum_{a\in\mathcal{A}}\Delta\mu_{aijs} + \begin{cases} \frac{Q}{NRMSE_s^{Min}}, & \text{if } (i,j) \text{ in } NRMSE_s^{Min} \text{ path} \\ 0 \quad , & \text{otherwise} \end{cases} \tag{7.7}$$

where $\rho \in [0, 1)$ is the pheromone evaporation rate, Q is the pheromone amount carried by every ant and $\mathcal{A}$ is the set of ants, $\mathcal{A} = \{1, 2, ..., A\}$. An additional $\frac{Q}{NRMSE_s^{Min}}$ pheromone change is given to edges included in the ant path with the minimum NRMSE value on segment s.

After the offline pheromone update, the ant paths are checked for energy feasibility in line 20 in Algorithm 5 and only energy feasible paths are returned from the $getEnergyFeasiblePaths$ method. Then the set of best paths is updated to include ant paths from the current iteration if they are better than the current best paths, ensuring that the total number of paths in $bestPaths$ does not exceed P. The $updateBestPaths$ method adds a path b from $possibleBestPaths$ to $bestPaths$ if b Pareto dominates a path in $bestPaths$, or if b is not dominated by any path in $bestPaths$ *and* has a better NRMSE value than a path in $bestPaths$. In the case of ACO for initial parent population generation, the $bestPaths$ set contains the P best paths for *each* segment, i.e. $|\mathcal{S}_u| \cdot P$ paths in total. When ACO is used in mutating crossover on the other hand, only P paths are included in the $bestPaths$ set, since all ants traverse the same segment in this case. After possibly updating the set of best paths, a new iteration of the overall for-loop in lines 4-22 begins. After I iterations, the best paths on each segment must be combined if ACO has been used for initial population generation, i.e. all the starting vertices of the ants are not the same. This is done in the $combineSegmentPaths$ method, which creates all combinations of the best paths for each segment into full paths and then selects P non-dominated and energy feasible full paths. Finally, the algorithm returns the set of paths in $bestPaths$.

7.2 Exact Method for Evaluation

To evaluate the HEA, it is compared to an exact method called multi-objective Dijkstra's algorithm (MDA). This algorithm was used in our preparatory research project to generate the set of optimal paths, i.e. the Pareto front, for the static, single-UAV path planning (SUAVPP) problem. The SUAVPP problem has many similarities to the MUAVPP problem, but it only considers a single UAV and hence does not account for the collision avoidance constraints explained in Chapter 5. Furthermore, it uses the single-UAV perspective equations for the risk, visual pollution and noise pollution objectives described in Chapter 6. Thus, the evaluation of the HEA consists of comparing the runtimes and the Pareto fronts generated by the MDA for each UAV in a system to the results from the HEA with $isSystem = False$, to see whether the paths found by the non-system HEA are close to the optimal paths for individual UAVs. This investigation will again give insights into the quality of the HEA with $isSystem = True$, and is discussed in detail in Section 8.3.

Simply put, the MDA finds all Pareto optimal paths between two vertices in a graph. The MDA cannot be applied to the MUAVPP problem directly, and we thus begin by decomposing the problem into a set of SUAVPP problems by ignoring the collision avoidance constraints and using the single-UAV perspective models for risk, visual and noise. Each of these single-UAV problems have a primal block-angular structure. The primal block-angular structure means that further decomposition of the single-UAV problem into independent path planning problems for each *segment* of the UAV's voyage is hindered by a complicating constraint, i.e. a constraint that makes the subproblems dependent. It is the energy capacity constraint (5.12) that is complicating, since it applies across all segments of a UAV's voyage. Thus, we temporarily ignore the energy constraint, allowing for the decomposition of the single-UAV problem into one path planning *subproblem* per segment. Since each segment consists of a start and end vertex, the MDA can be applied to solve each of these subproblems.

After all subproblems have been solved using the MDA, the global Pareto front for every UAV u in the system is found by combining the Pareto fronts for every segment $s \in \mathcal{S}_u$ and removing paths that do not comply with the energy constraint and paths that are Pareto dominated. The described approach is evidently not needed if the voyage of UAV u only consists of one segment. By the principle of optimality in dynamic programming, the combination of Pareto fronts from independent subproblems will lead to the global Pareto front for the original problem. However, when a complicating constraint is introduced, the combination approach described could fail to generate the entire global Pareto front. To see this, consider the simple example with a voyage consisting of two segments. On the first segment, a segment path p_1 is Pareto dominated by another segment path p_2. Thus, p_1 is discarded. When combining the Pareto fronts to find the global front, the path using segment path p_2 on the first segment is found to break the complicating constraint. If p_1 was *only* Pareto dominated by p_2 and the path consisting of segment path p_1 instead of p_2 complies with the complicating constraint, this path is missing from the Pareto front. Thus, the approach has not succeeded in generating the full, global Pareto front. However, due to the strong correlation between the energy consumption model proposed in Section 6.2 and the flight time, this situation is rather unlikely.

7.2.1 Multi-Objective Dijkstra's Algorithm

The MDA used to find Pareto optimal paths for a UAV u on segment $s \in \mathcal{S}_u$ is heavily based on Maristany de las Casas et al. (2021). The MDA follows the same principles as the single-objective version of Dijkstra's algorithm (Dijkstra, 1959). The MDA is a label-setting algorithm that applies Pareto dominance on possible paths in the graph, represented as *labels*. Given a path ending in vertex v, the unique label representing the path is given by $z = (v, c(z), e(z), p(z))$. $c(z)$ is the cost vector of the path, and corresponds to a vector of aggregated single-UAV perspective objective values. $e(z)$ is the aggregated energy consumption along the path, and $p(z)$ is the path traversed from the start vertex b to v, represented as a list of vertices. Using the definition of Pareto dominance described in Subsection 3.3.1, we say that a label $z = (v, c(z), e(z), p(z))$ Pareto dominates another label $k = (w, c(k), e(k), p(k))$, and we write $z \prec k$, if and only if

$$v = w \qquad and \qquad c(z) \prec c(k)$$

The energy consumption of each of the labels, i.e. $e(z)$ and $e(k)$, are not included in the dominance test, since energy consumption is a constraint and not an objective. The MDA finds the *minimum complete set* of the efficient, i.e. Pareto optimal, paths. A minimum complete set of efficient paths implies that there exists exactly one efficient path for every non-dominated cost vector. This means that if several paths have the exact same cost vectors, only the first one to be found by the MDA is included in the set of efficient paths. Thus, cost vector *equivalence* is enough to discard a label. The MDA uses a dynamic programming approach, where it traverses the graph by extending previously explored partial paths to find all Pareto optimal paths from the start location to the end location on a given segment of a UAV u. The MDA can be applied because the principle of optimality in dynamic programming holds, since all objectives are non-negative, which is a requirement for the principle of optimality.

In short, the MDA will, for a given graph $G = (\mathcal{V}, \mathcal{E})$ and given start and end vertices b and f of a segment of a UAV's voyage, find the efficient label set $\mathcal{Z}_f$, i.e. the efficient labels of the end vertex f. The MDA is shown in Algorithm 6. The algorithm starts by initializing several sets used throughout the algorithm, in lines 2-7. $\mathcal{K}$ is a priority queue, which consists of *candidate labels* for every vertex, i.e. labels that are next to be explored in the algorithm. Each vertex v has at most one associated label in $\mathcal{K}$, and this is the lexicographically smallest candidate label for the vertex. In addition, the MDA manages a set of efficient labels $\mathcal{Z}_v$ for every vertex v. The total set of efficient labels $\mathcal{Z}$ is the union of all the efficient label sets for all vertices, as shown in line 7. $\mathcal{D}$ is a set indexed by the edges in the graph, containing an index for the last processed label for every edge.

In lines 8-9, a label for the start vertex b is created and added to the priority queue $\mathcal{K}$. The rest of the algorithm consists of the while-loop in lines 10-23. The MDA continues until there are no labels left in the priority queue, meaning there are no more candidate labels that may dominate the labels in the efficient label set for the end vertex f. In line 11, the next label to

Algorithm 6 Multi-objective Dijkstra's algorithm used to generate the set of Pareto optimal paths for a UAV u on segment s

```
    Input: Graph $G = (\mathcal{V}, \mathcal{E})$, start vertex $b$, end vertex $f$, set of lower bound cost vectors $\underline{\mathcal{C}}$
    Output: Set of Pareto optimal labels for the end vertex $f$, $\mathcal{Z}_f$
 1: function MULTOBJECTIVEDIJKSTRA($G$, $b$, $f$, $\underline{\mathcal{C}}$)
 2:     $\mathcal{K} \leftarrow \emptyset$
 3:     $\mathcal{D} \leftarrow \emptyset$
 4:     for $v \in \mathcal{V}$ do
 5:         $\mathcal{Z}_v \leftarrow \emptyset$
 6:     end for
 7:     $\mathcal{Z} \leftarrow \bigcup_{v \in \mathcal{V}} \mathcal{Z}_v$
 8:     $z_b \leftarrow (b,\ (0,\ ...,0),\ 0,\ None)$
 9:     $\mathcal{K} \leftarrow \mathcal{K}.insert(z_b)$
10:     while $\mathcal{K} \neq \emptyset$ do
11:         $z_v \leftarrow extractLexicographicallyMin(\mathcal{K})$
12:         $\mathcal{Z}_v \leftarrow \mathcal{Z}_v.insert(z_v)$
13:         $z_v^c \leftarrow getNextCandidateLabel(v,\ \mathcal{D},\ \mathcal{Z},\ \underline{\mathcal{C}})$
14:         if $z_v^c$ is not $None$ then
15:             $\mathcal{K} \leftarrow \mathcal{K}.insert(z_v^c)$
16:         end if
17:         $v_{prev} \leftarrow getLastVisitedVertex(z_v)$
18:         for $w \in getNeighbors(v)$ do
19:             if $w$ is not $v_{prev}$ then
20:                 $\mathcal{K} \leftarrow propagate(G,\ z_v,\ w,\ \mathcal{K},\ \underline{\mathcal{C}})$
21:             end if
22:         end for
23:     end while
24:     return $\mathcal{Z}_f$
25: end function
```

be explored is extracted. The selected label is the *lexicographically* smallest label z_v in the priority queue $\mathcal{K}$. The exploration strategy of the MDA is hence dependent on the order of the costs in the cost vector. In this thesis, the order of the costs is flight time, risk, visual pollution and noise pollution. Thus, until the algorithm has found an efficient label for the end vertex f, the graph is explored in a breadth-first manner, following the flight time cost of every label. The extracted label z_v is added to the efficient label set $\mathcal{Z}_v$ in line 12, since it is guaranteed to be a non-dominated label for vertex v.

The next step in the algorithm, shown in lines 13-16, is to search for a new candidate label for the vertex v corresponding to the extracted label z_v using the $getNextCandidateLabel$ method. This method works by finding the lexicographically smallest efficient label, z_w, where w is a neighbor of v, and creating a new candidate label for v by extending the label z_w. A more in-depth description of the $getNextCandidateLabel$ method is found in Appendix C. If a candidate label is found, it is added to the priority queue.

After using the $getNextCandidateLabel$ method to extend labels along edges going *in* to vertex v, the MDA extends labels along the edges going *out* from vertex v using the $propagate$ method in lines 17-22. The $propagate$ method extends the label z_v along an outgoing edge (v, w), and potentially updates the priority queue for vertex w. The method is described in detail in Appendix C. To avoid unnecessary computations, the algorithm does not propagate to the vertex v_{prev} that it just arrived from, as this extension will never lead to a non-dominated label in v_{prev}.

The $getNextCandidateLabel$ and $propagate$ methods both apply the lower bound pruning technique proposed by Maristany de las Casas et al. (2021) to decrease the search space of the MDA. This technique uses a set of *lower bound cost vectors*, which consist of a vector of the optimal cost for every objective to get from every vertex v in the graph to the end vertex f. To generate the set of lower bound cost vectors, *single-objective* MDA is used to solve the shortest path problem from the end vertex f to every other vertex in the graph, for every objective. These lower bound cost vectors are then used to check whether it is necessary to extend a path ending in v to vertex w. The lower bound cost vector for vertex w, i.e. the lowest value for each objective for a path from w to end vertex f, is added to the current cost vector in the label for vertex v along with the cost of going from vertex v to w. If this total cost vector is Pareto dominated by or equal to the cost vector of a label in the efficient label set for end vertex f, there is no need to extend the path from v to w, as the path will never be added to the set of efficient labels for f. In addition to the lower bound pruning, the $getNextCandidateLabel$ and the $propagate$ methods perform two other Pareto dominance checks. The new candidate label or propagated label for vertex v, z_v, must not be dominated by or have equal cost vector as the efficient label set of v or the efficient label set of the end vertex f. Furthermore, the label z_v must comply with the energy consumption constraint.

Chapter 8

Computational Study

This chapter presents the results of applying the hybrid evolutionary algorithm (HEA) explained in Chapter 7 to the static, multi-UAV path planning (MUAVPP) problem. In this chapter, we introduce the terms $\mathrm{HEA}^{\mathrm{C}}$, $\mathrm{HEA}^{\mathrm{T}}$ and $\mathrm{HEA}^{\mathrm{F}}$. $\mathrm{HEA}^{\mathrm{C}}$ denotes the complete algorithm, i.e. running HEA with $isSystem = True$, which recursively calls HEA with $isSystem = False$ once for each UAV in the system to create the initial population. Furthermore, $\mathrm{HEA}^{\mathrm{F}}$ refers to running HEA with $isSystem = False$, while $\mathrm{HEA}^{\mathrm{T}}$ refers to all other parts of the $\mathrm{HEA}^{\mathrm{C}}$ except for the recursive calls of $\mathrm{HEA}^{\mathrm{F}}$.

The chapter is split into ten sections. Section 8.1 concerns the input parameters used in this computational study, while Section 8.2 describes our hypotheses and the test instances we developed to evaluate them. Section 8.3 compares $\mathrm{HEA}^{\mathrm{F}}$ to the exact multi-objective Dijkstra's algorithm (MDA) presented in Section 7.2. Section 8.4 discusses differences between the single-UAV perspective and the system perspective, i.e. differences between the results from $\mathrm{HEA}^{\mathrm{F}}$ and $\mathrm{HEA}^{\mathrm{C}}$. In Section 8.5, the overarching runtime results for $\mathrm{HEA}^{\mathrm{C}}$ are presented, while Section 8.6 discusses the effects of various components, operators and parameters of the $\mathrm{HEA}^{\mathrm{C}}$. Furthermore, Section 8.7 analyzes the stability of $\mathrm{HEA}^{\mathrm{C}}$ and Section 8.8 concerns the impact of the ground square size. Section 8.9 investigates the system behavior when running $\mathrm{HEA}^{\mathrm{C}}$ and evaluates each of the hypotheses. Finally, Section 8.10 summarizes the findings in regards to the hypotheses and takes a broader view on the results presented in this computational study.

The code for this thesis is written in Python because of its popularity, accessibility and our prior experience with the programming language. All the test instances are run on Lenovo M5 servers with two 3.4 GHz Intel E5-2643v3 processors, 512 GB RAM, 12 cores and 12 threads.

8.1 Input

This computational study makes use of several parameters regarding the urban environment, the UAVs, the objective models and the solution method. The values for these parameters are presented and explained in Appendix D. The computational study uses the Norwegian city of Stavanger as a case study. The area used is shown in Figure 8.1, and spans an area of 4.97 km west-east, referred to as the x-direction in this thesis, and 2.60 km south-north, referred to as the y-direction. Subsection 8.1.1 presents a subset of the input parameters that are tested with various values in this computational study.

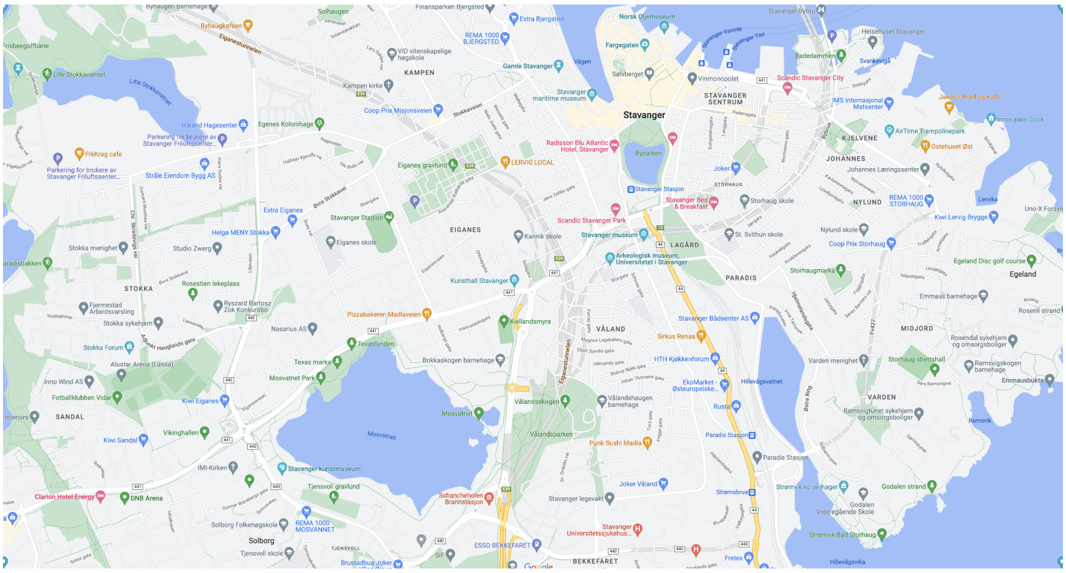


Figure 8.1: The urban environment used in the computational study (GoogleMaps, 2023).

8.1.1 Modified Parameters

Table 8.1 gives an overview of the parameters that are tested with various values in the computational study, which we refer to as *modified parameters*. The parameters above the horizontal line belong to the world and objective models, while those below the line are solution method parameters. Each parameter is given a base value, as well as a reduced and an increased value. In the following sections, the parameters take their base values unless it is stated otherwise. The term *base case* is used to denote the case where all parameters take their base values.

Modified World and Objective Model Parameters

Seven world and objective model parameters are tested with different values in the computational study. What values these parameters will get in a future with widespread UAV usage is to a large extent a political question, and thus up to policy makers. We therefore want to test different values, to investigate which parameters the system flight plans returned from the

Table 8.1: Overview of the values of the modified parameters. Each modified parameter is given a base value, a reduced value and an increased value.

Parameter [unit]	Description	Base	Reduced	Increased
α	Factor for maximum reduction in mean time between falldowns due to the possibility of collisions	0.5	0.3	0.7
β	Discount factor for the system perspective visual pollution model	0.7	0.6	0.9
δ [s]	Length of time interval regarded as simultaneous UAV impact	5	3	7
λ	Ground square side length multiple of L	10	6	12
D^{Max} [s]	Collision risk threshold	45	30	60
R^N [dB]	Threshold for noise pollution	55	52	58
R^V [m]	Threshold for visual pollution	0.005	0.0025	0.0075
N	Population size for HEA$^{\text{T}}$	30	20	50
p^{Mut}	Threshold for performing a mutating crossover for HEA$^{\text{C}}$	0.8	0.5	0.9

HEA$^{\text{C}}$ are sensitive to and thus must be carefully assessed by policy makers. For more details regarding each of the parameters, refer to Chapter 6.

The first parameter is α, for which we set the base value to 0.5. Thus, the mean time between falldowns can at most be halved compared to when no other UAVs are flying close by. The second parameter is β, which is given a base value of 0.7. This implies a significant increase in the visual pollution value from each additional UAV when the total number of UAVs is low, while simultaneously implying a clear diminishing effect as the number of UAVs increases. Third, we adjust δ, which both decides what is considered simultaneous impact for the visual and noise pollution models, and how often relative positions between UAVs are checked in the risk model. Based on our judgment, a base value for δ of 5 seconds is deemed reasonable. Fourth, the ground square multiple λ is adjusted. The base value is set to 10, making the side length of a ground square $\lambda \cdot L = 10 \cdot 10$ meters $= 100$ meters. The reduction and increase in λ correspond to a change in the number of $\lambda = 1$ ground square equivalents of 64 and 44, respectively, which we consider a reasonable range. Fifth, we adjust the collision risk threshold parameter D^{Max}, which decides the maximum distance between two UAVs that results in increased risk. We set the base value to be three times the value of D^{Min}, which is the minimum distance between two UAVs to avoid collision. D^{Min} is set to 15 meters because this ensures that two UAVs are at least an edge away from each other, as the length of a diagonal edge is $\sqrt{10^2 + 10^2} = 14.14$ meters. Hence, the base value for D^{Max} is 45 meters, and the collision risk is increased if UAVs are located within the area 30 meters outside the collision area. The sixth parameter is the threshold for noise pollution, R^N, which determines which ground squares that are affected by noise pollution. There exists various recommendations regarding noise pollution thresholds. We set the base value to 55 dB, which seems reasonable when looking at WHO recommendations for community noise exposure and guidelines from the Norwegian Government regarding outdoor noise (Aviation Environment Federation, n.d.; Klima- og miljødepartementet, 2021). Using the formulas in Section 6.5, we see that a 3 dB increase from the base value approximately corresponds to a 25% higher noise pollution value, which we think is a fair deviation when testing this parameter. Finally, we also test different values for the visual pollution threshold, R^V, which determines the ground squares affected by visual pollution. Current regulations have not set a threshold for visual pollution, and we thus set 0.005 meters as the base value based on our own judgment.

Modified Hybrid Evolutionary Algorithm Parameters

There are numerous possible combinations of parameter values that can be tested for the solution method. Due to time constraints, we have limited the adjustments to the parameters we believe are most relevant. The parameters we adjust concern the NSGA-III, and not the ACO. The reason for this is two-fold; NSGA-III is our main algorithm, and the parameters in NSGA-III are more independent of each other, which is explained in Appendix D, making it more appropriate to look at the effects of changing single parameters in isolation.

The first algorithmic parameter we adjust is the population size, N, for HEA$^{\mathrm{T}}$. There are some variations across different sources regarding what population size is appropriate. We believe that 30 is a reasonable base value for N, for two main reasons. First, preliminary testing showed that the HEA$^{\mathrm{C}}$ has long runtimes, and that the runtime has a strong, positive correlation to population size. Thus, a smaller population size is necessary, despite population sizes of up to 100 individuals being common. Second, the use of HEA$^{\mathrm{F}}$ within the HEA$^{\mathrm{C}}$ to improve the initial parent population allows for a more guided search for individuals, and can thus help to make up for the smaller population. To further test whether 30 is a fair population size, we also reduce the HEA$^{\mathrm{T}}$ population down to 20 to see if an even smaller population could produce the same results. Similarly, we test a HEA$^{\mathrm{T}}$ population size of 50, to see if this leads to large enough improvements to make up for the additional complexity. The second algorithmic parameter is the threshold for performing a mutating crossover in both HEA$^{\mathrm{F}}$ and HEA$^{\mathrm{T}}$, p^{Mut}. We believe 0.8 is a fair base value. This implies that mutating crossover is used in the creation of one out of every five offspring on average, i.e. new paths are introduced regularly although regular crossover is most prevalent. We set the increased value to 0.9, since this implies a halving in the number of times mutating crossover occurs. For the reduced value, we test with 0.5 since this implies that both crossover operators are equally likely.

8.2 Hypotheses and Test Instances

This section explains the main areas investigated in this computational study. Subsection 8.2.1 presents our hypotheses about the system flight plans generated by the HEA^{C} when applied to the static MUAVPP problem. Based on the hypotheses, we have developed a set of test instances, which are explained in Subsection 8.2.2.

8.2.1 Hypotheses

In our preparatory research project, we studied how a single UAV behaved in an urban environment. Our focus in this computational study is to investigate how the introduction of a system of UAVs affects UAV behavior. We have developed a set of eight hypotheses to guide the computational study, which are summarized in Table 8.2. The first four hypotheses concern the general interaction between UAVs and the effect of the environment on system behavior. The first hypothesis, H1, is that population density trumps sheltering factor, i.e. UAVs will not want to fly above areas with a large population even if the sheltering factor is high. The results from our preparatory research project indicated that this hypothesis holds in the single-UAV setting. We therefore want to test whether it still holds for the multi-UAV setting, thus indicating that environment characteristics affecting individual UAVs are not neglected when introducing a system of UAVs. The second hypothesis, H2, investigates this area further by testing whether system effects might be considered *more important* than environment characteristics, given that the characteristics are not neglected. More precisely, H2 states that a UAV flying far away from other UAVs in a system behaves differently when another UAV is introduced in its vicinity. Such an introduction can directly influence the behavior of the previously lonely UAV through the need to consider collision avoidance, the other UAV's collision risk area and the possibility to affect the same ground squares in regards to visual and noise pollution. The third hypothesis, H3, is that UAVs with a lower importance of efficiency are the ones responsible for collision avoidance when encountering UAVs with higher importance of efficiency. By responsible for collision avoidance, we refer to a UAV making a deviation from its path to fly either above, below or around the path taken by another UAV to avoid a collision. As explained in Section 7.1, only one UAV changes its path when two UAVs are on collision course. The fourth hypothesis, H4, builds on H3 and states that given equal importance of efficiency values, UAV characteristics determine which UAV is responsible for collision avoidance when two UAVs meet. Given the system perspective objective models, a UAV with a higher weight results in a higher risk value, a UAV with a larger size results in a higher visual pollution value, and a

UAV producing more noise results in increased noise pollution. It thus seems reasonable to assume that UAVs with lower values for such UAV characteristics are responsible for collision avoidance, since their overall impact on the system perspective objectives are smaller than the impact of a UAV with higher values for the UAV characteristics. Since only two UAV types are used in this thesis, H4 is equivalent to delivery UAVs being responsible for collision avoidance when encountering passenger UAVs.

The last four hypotheses concern modified parameters described in detail in Subsection 8.1.1, and focus on flight proximity between UAVs, i.e. the degree to which UAVs fly close over a period of time. Hypotheses H5-H7, state that α, β and δ, respectively, affect the proximity of UAVs. Similarly, the last hypothesis, H8, is that the thresholds D^{Max}, R^V and R^N affect the proximity of UAVs.

Table 8.2: Overview of hypotheses.

Hypothesis	Description
H1	Population density trumps the sheltering factor, i.e. UAVs will not want to fly above areas with a large population even if the sheltering factor is high
H2	A UAV flying far away from other UAVs in a system behaves differently when another UAV is introduced in its vicinity
H3	UAVs with a lower importance of efficiency are responsible for collision avoidance when encountering UAVs with higher importance of efficiency
H4	Given equal importance of efficiency values, UAV characteristics determine which UAV is responsible for collision avoidance when two UAVs meet
H5	The maximum reduction in mean time between falldowns due to the possibility of collisions, α, affects the proximity of UAVs
H6	The visual pollution discount factor, β, affects the proximity of UAVs
H7	The length of a time interval, δ, affects the proximity of UAVs
H8	The thresholds for risk, visual pollution and noise pollution, i.e. D^{Max}, R^V and R^N, affect the proximity of UAVs

8.2.2 Test Instances

We have developed a set of three test instances to evaluate the eight hypotheses; TI1, TI2 and TI3. All UAVs are one of two types, either a passenger UAV based on the EHang 216 Autonomous Aerial Vehicle or a delivery UAV based on the Amazon Prime Air UAV. Refer to Appendix D for further details regarding the two UAV types. The distances between the locations for each UAV in the three test instances have been kept rather short, due to the computational complexity of the system perspective objective models. Since there are many UAVs in each test instance, we have chosen not to assign a specific story to the voyage taken by each UAV.

Instead, we have assigned varying payloads, locations and importance of efficiency values so that the UAVs cover a wide variety of use cases. Typical non-urgent commercial use cases can be the delivery of clothes or leisure articles bought online. More urgent commercial use cases include the delivery of warm takeout food or UAV taxi services for passengers in a hurry. Medical use cases can also be more or less urgent. Prominent urgent medical use cases include transport of healthcare personnel or medical supplies in case of emergency. Examples of non-urgent medical use cases can be routine deliveries of supplies to hospitals, or transport of medical personnel to and from planned home visits. The voyage of a UAV will often have varying importance of efficiency values for the different segments. Despite a delivery or passenger transport being urgent, the return trip will often not be, thus resulting in a low importance of efficiency on the return segment. Another example where the importance of efficiency can vary between segments is for vaccine delivery. Consider a vaccine that deteriorates if kept out of the fridge for too long. A UAV that first flies to a hospital to pick up these vaccines and then transports the vaccines to a local doctor's office, can have low importance of efficiency on the first segment and a high importance of efficiency on the second segment.

Each of the thee test instances are explained below. For all the test instances, the payload weights for the passenger UAVs are based on the average weight of Norwegian males at 86.6 kg (FHI, 2021). Furthermore, we have chosen to let the importance of efficiency parameter, J_{us}, take the values 1, 2 or 5. These values were chosen to ensure a notable difference in the importance of efficiency between a non-urgent, a somewhat urgent and an emergency segment of a UAV's voyage. The start times, T_u^S, for each UAV is set with the intent of creating possibilities for the UAVs to fly in close proximity and perform collision avoidance, to lay a good foundation for evaluating the hypotheses. Note that since hypotheses H5-H8 consider a UAV system as a whole, the testing of these hypotheses are not directly linked to certain UAVs in the test instances.

Test Instance 1

The ten UAVs in TI1 form clusters in three different areas in the environment. Figure 8.2 displays the voyages of the UAVs, with a straight line drawn for each segment. UAV 1.6 and 1.9 have locations on each side of the border between areas varying in terms of population and sheltering and are thus aimed at testing H1. Furthermore, the UAVs with crossing segments in the two clusters containing UAVs 1.1-1.5 and 1.6-1.9 are used to test H3 and H4. Moreover, UAV 1.10 is used to test H2, together with UAVs from TI2 and TI3. Table 8.3 shows different voyage-specific parameters for each of the UAVs in TI1.

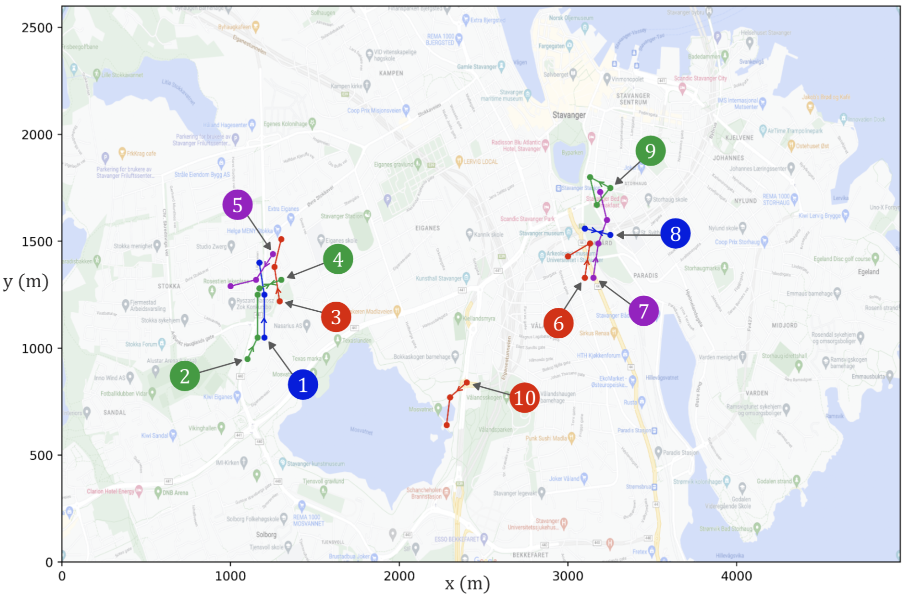


Figure 8.2: Illustration of the voyages for the UAVs in TI1. The “1.x” formulation of UAV names is omitted for simplicity, and each number points at the corresponding start location for each UAV. A straight line is drawn between the locations for each segment. Red and purple dots and lines represent delivery UAVs, while blue and green dots and lines represent passenger UAVs.

Table 8.3: Overview of voyage-specific parameters for each UAV in the system for TI1.

Test instance 1

UAV	Type	Start time, T_u^S [s]	Number of segments, $\lvert\mathcal{S}_u\rvert$	Importance of efficiency, J_{us}	Number of passengers, K_{us}	Payload weight, W_{us}^P [kg]
1.1	P	35	2	1, 1	0, 1	0, 86.6
1.2	P	1	2	1, 1	0, 2	0, 173.2
1.3	D	61	2	1, 1	-	1, 0
1.4	P	68	2	5, 1	1, 0	86.6, 0
1.5	D	68	2	1, 1	-	1, 2
1.6	D	2	2	1, 1	-	2, 1
1.7	D	2	3	1, 1, 1	-	2, 1, 0
1.8	P	46	2	1, 1	1, 2	86.6, 173.2
1.9	P	16	3	1, 1, 1	0, 1, 0	0, 86.6, 0
1.10	D	12	2	1, 1	-	1, 0

In the "Type" column we have P: Passenger, D: Delivery.

Test Instance 2

The ten UAVs in TI2 form clusters in four different areas in the environment. Figure 8.3 displays the voyages of the UAVs, as well as four additional fictional no-flight zones. As explained in Section 2.3, authorities can introduce new no-flight zones, and we consider the introduction of additional no-flight zones in urban areas plausible as the UAV service market grows. UAVs 2.4 and 2.5 have their first segments crossing, and are designed to test hypothesis H3. Furthermore, UAV 2.10 is equal to UAV 1.10 in TI1, but in contrast to UAV 1.10, UAV 2.10 has another UAV, namely UAV 2.9, flying close by. These UAVs are thus aimed at testing H2, i.e. the behavior of system flight plans when a UAV flies “by itself” in a system compared to when another UAV is flying close by. Table 8.4 shows the different parameters for the UAVs.

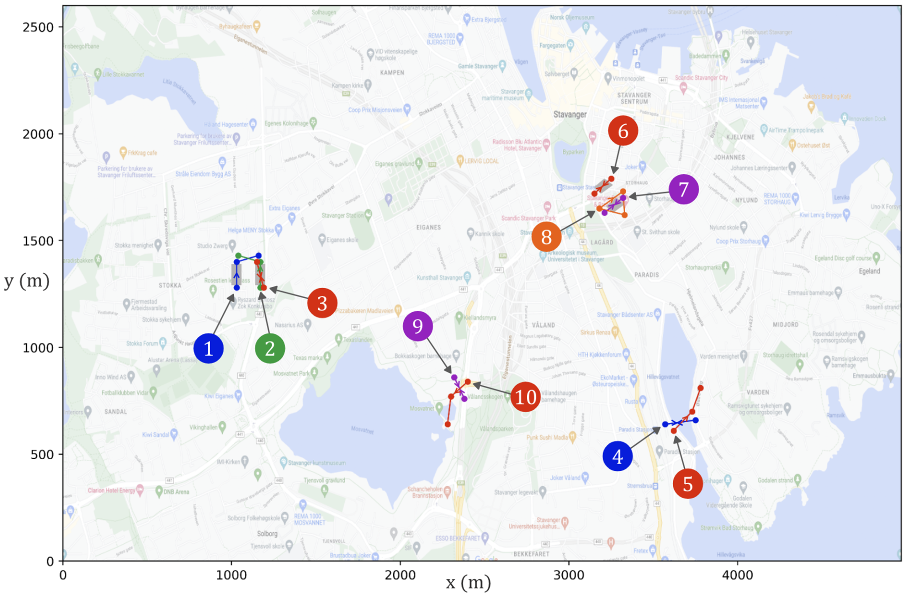


Figure 8.3: Illustration of the voyages for the UAVs in TI2 and TI3. The “2.x”/"3.x" formulation of UAV names is omitted for simplicity, and each number points at the corresponding start location for each UAV. A straight line is drawn between the locations for each segment. Red, orange and purple dots and lines represent delivery UAVs, while blue and green dots and lines represent passenger UAVs. The four light gray polygons correspond to the fictional no-flight zones in TI2.

Test Instance 3

TI3 is identical to TI2 except for the absence of the fictional no-flight zones and changed importance of efficiency values for UAVs 3.4 and 3.5 compared to UAVs 2.4 and 2.5. By testing both with and without the fictional no-flight zones, we will see the effect of such obstruction more clearly. The changed importance of efficiency values for UAVs 3.4 and 3.5 are meant to explore H3 in conjunction with UAVs 2.4 and 2.5 in TI2. Furthermore, UAV 3.9 and 3.10 are used to test H2. Table 8.5 shows the parameter values that are changed for TI3 compared to TI2.

Table 8.4: Overview of voyage-specific parameters for each UAV in the system for TI2.

Test instance 2

UAV	Type	Start time, T_u^S [s]	Number of segments, $\lvert\mathcal{S}_u\rvert$	Importance of efficiency, J_{us}	Number of passengers, K_{us}	Payload weight, W_{us}^P [kg]
2.1	P	5	2	1, 1	1, 0	86.6, 0
2.2	P	5	2	2, 1	2, 0	173.2, 0
2.3	D	3	2	2, 1	-	1, 0
2.4	P	5	2	5, 1	1, 0	86.6, 0
2.5	D	4	2	1, 1	-	2, 0
2.6	D	11	2	2, 1	-	1, 0
2.7	D	10	2	1, 1	-	1, 0
2.8	D	11	3	1, 1, 1	-	2, 1, 0
2.9	D	11	2	1, 1	-	1, 0
2.10	D	12	2	1, 1	-	1, 0

In the "Type" column we have P: Passenger, D: Delivery.

Table 8.5: Overview of voyage-specific parameter changes for TI3 compared to TI2.

Test instance 3

UAV	Importance of efficiency, J_{us}
3.4	1, 1
3.5	5, 1

8.3 Exact Method Comparison

This section compares the HEA to the exact MDA method. As explained in Section 7.2, the MDA solves the single-UAV path planning problem instead of the more complex multi-UAV path planning problem. Hence, HEA^F, which also finds paths for a single UAV, is used for the comparison instead of HEA^C. The suitability of a solution method is a balance between runtime and solution quality, and we therefore look into both of these aspects. An important thing to note about both this and remaining sections of the computational study, is that the short distances between UAV locations in the test instances only make our findings indicative, and should thus be considered as a sign that further research should be conducted.

Table 8.6 presents the results from running HEA^F and MDA on TI1-TI3 with base values for the modified parameters. The solution methods are run for each UAV in parallel, and thus the numbers shown in the table reflect the longest runtime among the UAVs. It is clear from the table that the HEA^F has much shorter runtimes than the MDA. Whereas the HEA^F values range from

Table 8.6: Runtimes of HEA^F and MDA for each test instance. The solution methods have been run in parallel for each UAV, and the runtimes correspond to the longest runtime among the UAVs in the system.

	Runtime [hrs]		
Solution method	**TI1**	**TI2**	**TI3**
HEA^F	18.1	11.5	9.1
MDA	302.6	207.0	185.9

9.1-18.1 hours, the MDA runtime stretches over 185.9-302.6 hours. We also see that for both solution methods, TI3 has the shortest runtime and TI1 has the longest runtime. TI1 is more complex than TI2 and TI3 since it contains more passenger UAVs, leading to more affected ground squares to calculate objective values for and thus increased runtime. The increased runtime of TI1 compared to TI2 and TI3 is discussed in depth in Section 8.5. The considerably longer runtime for the MDA for TI1 compared to TI2 and TI3, thus indicates that the runtime weakness of the MDA will only become more prominent as test instances increase in complexity. It should be noted that HEA^F is subject to randomness, and runtimes may hence vary when performing multiple runs. Running HEA^F ten times for TI1 resulted in runtime deviations of up to 15% of the average runtime value. The HEA^F runtimes are thus still considerably shorter than the MDA runtimes when accounting for the runtime variations. Efficient computations are important in UAV path planning, and the heuristic HEA hence seems like a more appropriate solution method than the exact MDA from a runtime perspective.

To gain insight into the solution quality of HEA^F, we compare the paths and corresponding objective fronts returned from the two methods. Due to the large number of UAVs in the test instances, we show the results for a representative set of UAVs when comparing the HEA^F to the MDA. Overall, the comparison gives three main insights, and we thus use a set of three UAVs to exemplify each of the conclusions.

Figure 8.4 shows three dimensional plots of the paths found by the HEA^F and the MDA for UAVs 1.6, 2.8 and 3.3. The figure shows that the HEA^F paths tend to be positioned within the large set of optimal MDA paths. Extreme changes of the environment characteristics between ground squares close to each other are rare, making it unlikely that HEA^F paths close to MDA paths at the same altitude will have very large differences in objective values. The results hence indicate that the HEA^F finds paths that are close to or at optimality. While the HEA^F in parts a) and b) in Figure 8.4 finds a set of paths that appear to include various path variations found by the MDA, the HEA^F paths found in Figure 8.4 c) only seem to include paths close to the MDA paths at low altitudes.

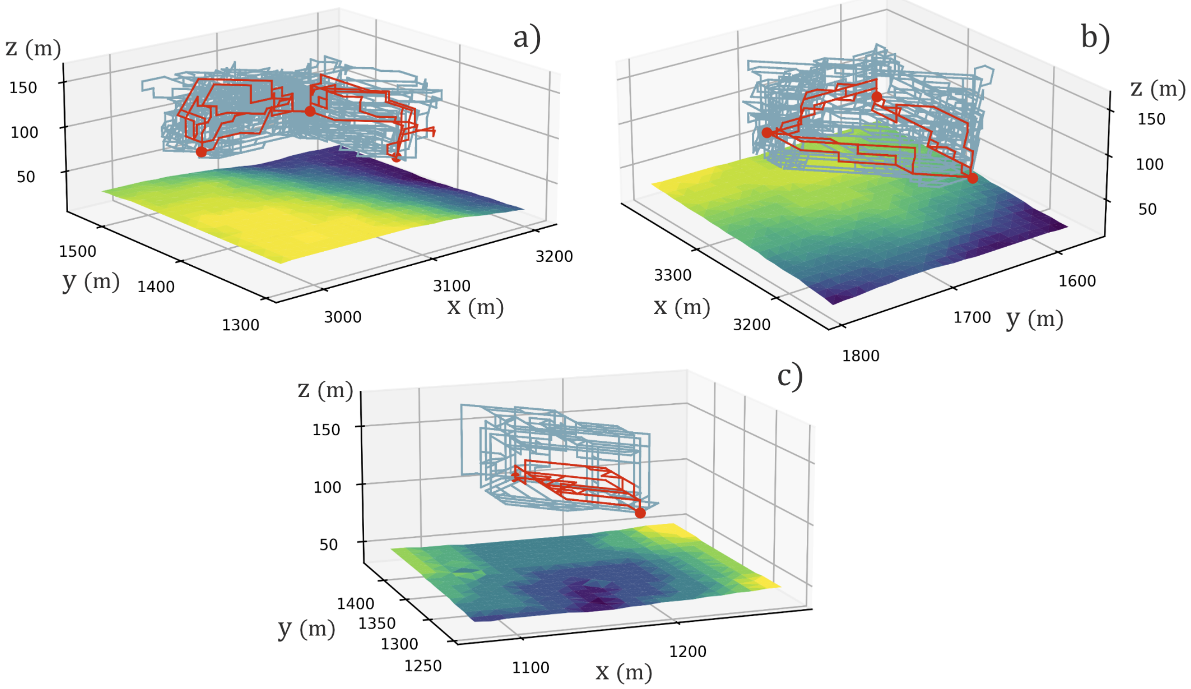
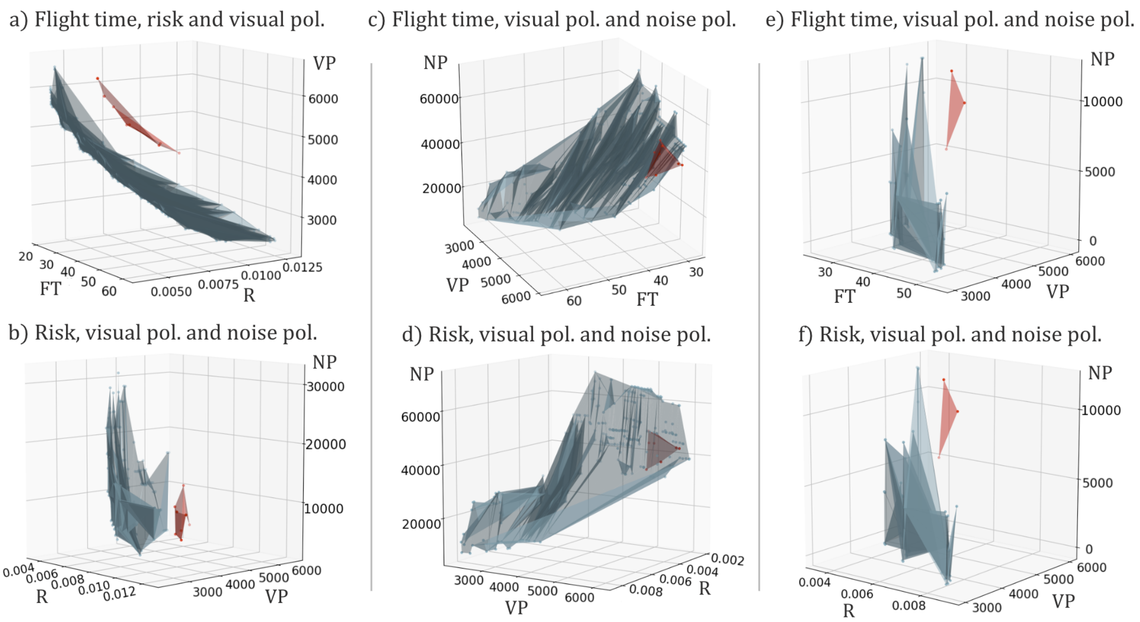


Figure 8.4: Illustration of three dimensional path plots for HEA^F and MDA. Part a) shows UAV 1.6, part b) shows UAV 2.8 and part c) shows UAV 3.3. The MDA paths are shown in blue, and the HEA^F paths are shown in red. Note that each part of the figure is angled differently to give the most representative view of the paths shown. Also note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

It is not sufficient to only compare the paths found by the two methods to assess whether the HEA^F finds optimal or close-to-optimal paths. Figure 8.5 shows three dimensional front plots for different combinations of the four objectives for the same three UAVs displayed in Figure 8.4. It should be noted that since the objective function vector is four dimensional, these three dimensional plots are only approximations and not exact fronts. Furthermore, there is no limit on the variation of the non-plotted objective. This is due to the number of single-UAV paths returned from HEA^F being small, and the probability of having the exact same or very close values for the non-plotted objective is thus miniscule. Figure 8.5 shows that HEA^F does not find the MDA paths with the more extreme trade-offs between the objectives. In general, it is important to return diverse paths so that a decision maker has a real choice regarding what social impact a system flight plan should have. Yet, it is unlikely that a decision maker prefers paths with very extreme trade-offs, since all four objectives are important in UAV path planning. Thus, the paths returned from HEA^F can still constitute a satisfactory degree of diversity. Figure 8.5 also shows that the HEA^F fronts generally seem to be positioned quite



a) Flight time, risk and visual pol.

c) Flight time, visual pol. and noise pol.

e) Flight time, visual pol. and noise pol.

b) Risk, visual pol. and noise pol.

d) Risk, visual pol. and noise pol.

f) Risk, visual pol. and noise pol.

Figure 8.5: Illustration of three dimensional fronts for different objective combinations for HEA^F and MDA. Parts a) and b) belong to UAV 1.6, part c) and d) to UAV 2.8 and part e) and f) to UAV 3.3. The gray points belong to the MDA, and the red points belong to HEA^F. Note that each part is angled differently to give the most representative view of the fronts. Also note that the illustration does not show true fronts since the problem is four dimensional, and the fronts in the illustration are three dimensional. The objectives displayed in the different plots are shown on the associated axis and stated above each plot, with "FT", "R", "VP" and "NP" being short for flight time, risk, visual pollution and noise pollution, respectively. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

close to the MDA fronts, with the HEA^F fronts in parts c) and d) appearing to be on the MDA fronts. This hence appears to be in line with the insights from the path plots in Figure 8.4 and further strengthens the case for HEA^F. Bringing the focus to UAV 3.3 again, parts e) and f) in Figure 8.5 show that the MDA finds paths with lower values for visual and noise pollution compared to HEA^F. This coincides well with HEA^F not finding paths that are similar to the higher-altitude MDA paths in Figure 8.4 c), since the single-UAV perspective objective models in Chapter 6 are based on higher altitude paths resulting in lower visual and noise pollution values. Parts e) and f) in Figure 8.5 could thus indicate that the HEA^F has a preference for shorter paths with a lower flight time objective value.

To summarize, the MDA has much longer runtimes than HEA^F. As shown in Chapter 6, the system perspective objective models require substantially more complex objective computations than the single-UAV perspective models. Thus, the use of a heuristic solution method seems to be the natural choice for the MUAVPP problem. Furthermore, HEA^F seems to find quite good paths, despite being a heuristic solution method. More specifically, there are three main insights from the comparison. First, the HEA^F finds paths that appear to be close to optimal, as illustrated by UAV 1.6. Second, HEA^F appears to find some optimal paths, as illustrated by the fronts for UAV 2.8, where parts of the HEA^F fronts are on the MDA fronts. Third, HEA^F can be inclined towards finding shorter paths with low values for the flight time objective, as opposed to longer paths that can reduce noise and visual pollution, as illustrated by UAV 3.3.

We take the results from the HEA^F comparison as a positive sign for HEA^C. With single-UAV perspective equations, the HEA seems to successfully maneuver around in the solution space and return paths with adequate diversity. Although HEA^C also incorporates collision avoidance and uses system perspective objective equations, the main workings of the algorithm in itself stays the same. Hence, we assume the good qualities of HEA^F to also be applicable for HEA^C. The differences between the paths returned from HEA^F and HEA^C are explored in Section 8.4.

8.4 Implications of Perspective: Single versus System

With the previous section motivating the choice of a heuristic solution method, this section looks at how the choice of perspective affects the resulting UAV paths. By comparing the single-UAV perspective and the system perspective, we can take a stance regarding whether it is worth using the more computationally intensive system perspective.

Figure 8.6 shows HEA^C and HEA^F paths for a selection of UAVs that exemplifies important findings in the comparison of the two perspectives. Part a) of the figure shows that the two perspectives result in very similar paths for UAV 1.1. In contrast, there are big differences between the paths from the two perspectives for UAV 2.3, shown in part b) of the figure. HEA^F returns paths on both sides of the fictional no-flight zone, while HEA^C only uses paths on one side. Part c) of the figure shows the HEA^C paths for UAV 1.6. In the first segment, which is shown rightmost in the plot, the HEA^C paths are spread out, but they coincide on the second segment. In contrast, the HEA^F paths are very spread out on the second segment. Furthermore, UAV 1.10 has locations far away from other UAVs, and is hence not at collision risk. Still, part d) of Figure 8.6 shows that the paths returned from HEA^F and HEA^C differ. The reason for this is that the system perspective objectives are calculated for the system as a whole and that UAV 1.10 can still affect the same ground squares as other UAVs in other parts of the graph. Figure 8.6 hence shows that the paths from HEA^F and HEA^C do not always coincide.

The comparison of HEA^F and HEA^C indicates that the choice of perspective influences the paths taken by the UAVs. The use of system perspective models for the objectives thus seems

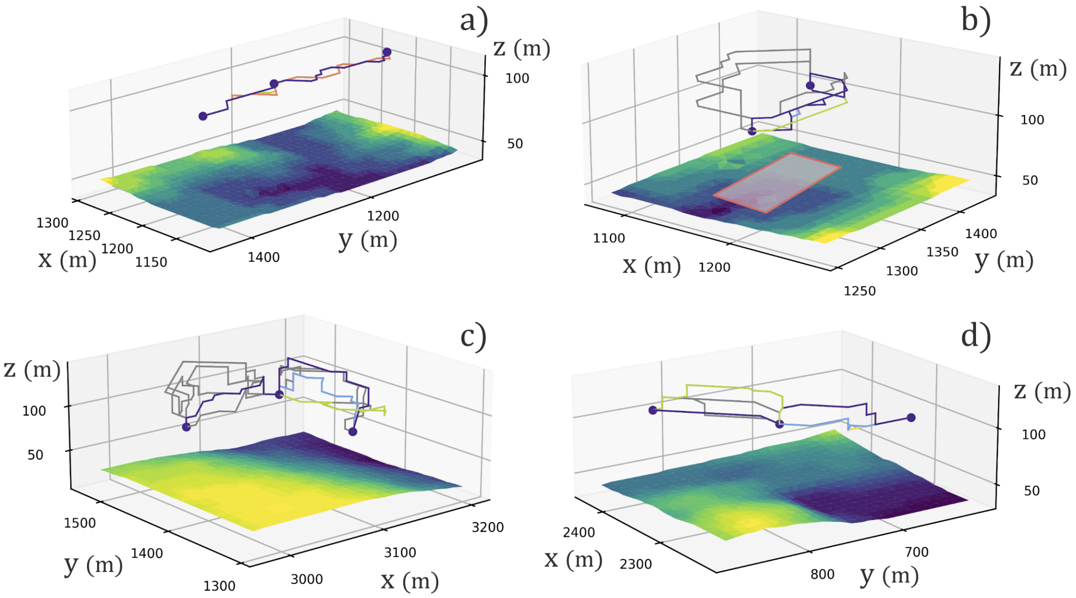


Figure 8.6: Illustration of three dimensional path plots for HEA^C and HEA^F. The different parts show UAVs as follows: a) 1.1, b) 2.3, c) 1.6 and d) 1.10. The HEA^F paths are shown in gray, and all other colors represent HEA^C paths. In cases where paths coincide, only one is displayed. If HEA^F and HEA^C paths coincide, a HEA^C path is displayed. Note that each part of the figure is angled differently to give the most representative view of the paths shown. Also note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level. The area framed in red on the ground in part b) corresponds to a no-flight zone.

to be valuable for finding good system flight plans. To further test this indication, we study the system perspective objective values of the initial population of HEA^C, i.e. system flight plans consisting of combinations of UAV paths from HEA^F, compared to the system perspective objective values of the final population in HEA^C. More precisely, we select five system flight plans from the initial population the same way that the five system flight plans returned from HEA^C are selected.

Figure 8.7 shows that the objective values of the five system flight plans from the initial and final populations of HEA^C are similar for the flight time objective, while all other objectives are considerably lower for the final population system flight plans. For risk, visual pollution and noise pollution, the maximum objective values of the five system flight plans from the final population are lower than the minimum values of the five system flight plans for the initial population. The even flight time values intuitively makes sense, since this objective is not directly affected by the system perspective. The objective models for risk, visual pollution

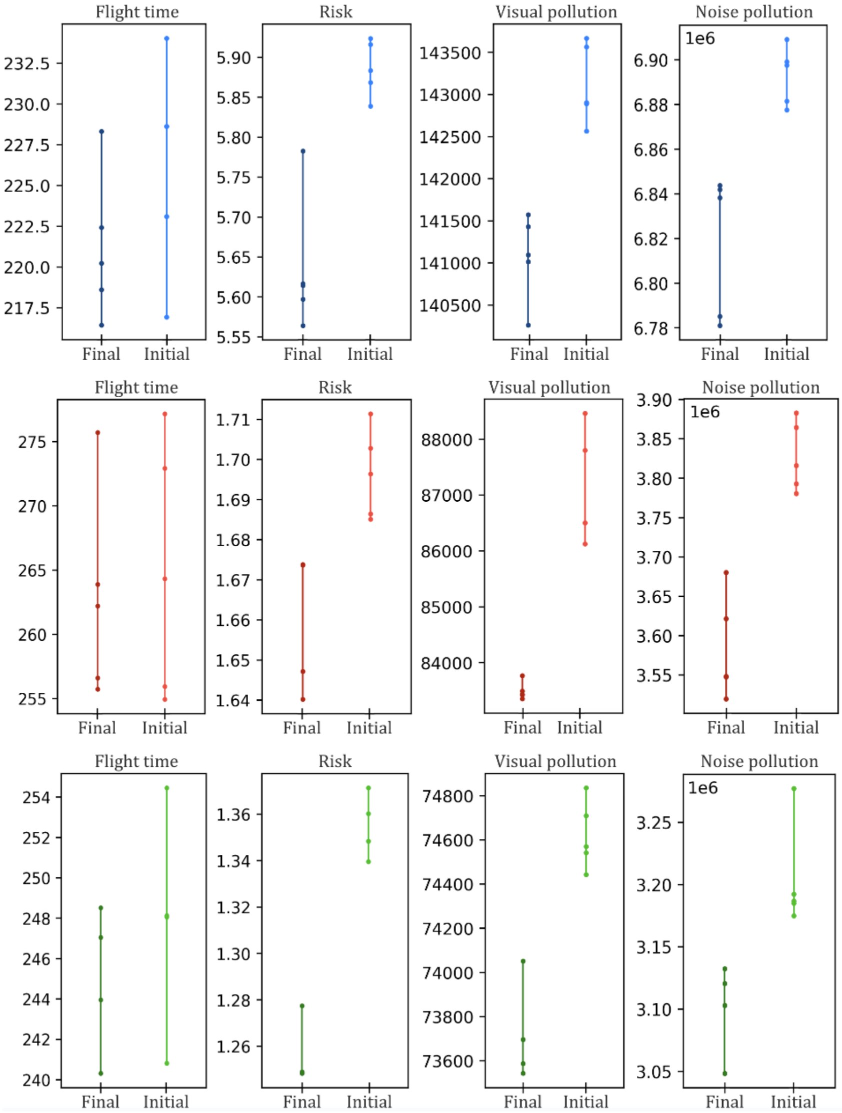


Figure 8.7: Illustration of the objective values of the five system flight plans selected from the initial and final population of HEA^C. The objective values of TI1, TI2 and TI3 are shown in the blue, red and green plots, respectively. The “1e6” in the corner of the noise pollution plots denotes that the values are in millions. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

and noise pollution are affected by the system perspective however, and the advantage of using the system perspective in the HEA^T part of HEA^C is thus further strengthened by the lower values for these objectives in the final system flight plans. We hence consider it worth taking the more computationally intensive system perspective when solving path planning problems regarding a system of UAVs.

8.5 Runtime Results

This section presents the runtime results for the modified parameter versions of HEA^C. To make the HEA^C versions as comparable as possible, every random occurrence in the code is seeded. This ensures that the same stream of random numbers is used for all versions of the HEA^C. The only exception is for a set of additional TI1 base case runs used in Section 8.6 to discuss the stability of the HEA^C. Tables 8.7 and 8.8 show the total runtimes for each test instance when modifying world and objective model parameters and algorithmic parameters, respectively.

Table 8.7: Runtimes of HEA^C for each test instance when modifying world and objective model parameters. Parameters with a "+" correspond to increased values compared to the base case (BC), while parameters with a "-" correspond to reduced values compared to the base case. The runtimes in each row below the "BC" row are given as multiples of the base case.

	Runtime		
Version	**TI1**	**TI2**	**TI3**
BC	185.8 hrs	132.3 hrs	113.4 hrs
α^+	0.972x	0.961x	0.966x
α^-	0.969x	0.960x	0.965x
β^+	0.968x	0.964x	0.967x
β^-	0.969x	0.963x	0.973x
δ^+	1.010x	0.973x	0.991x
δ^-	1.001x	0.975x	0.997x
λ^+	0.642x	0.577x	0.541x
λ^-	2.093x	2.041x	1.990x
D^{Max+}	0.978x	0.990x	0.992x
D^{Max-}	0.947x	0.955x	0.941x
R^{N+}	0.738x	0.718x	0.701x
R^{N-}	1.308x	1.319x	1.336x
R^{V+}	0.825x	0.804x	0.793x
R^{V-}	1.505x	1.524x	1.557x

Overall, Tables 8.7 and 8.8 show that the HEA^C has long runtimes. Preliminary profiling of the algorithm indicated that most aspects of HEA^C are in fact quite fast. However, the calculations of the system perspective objective values to generate fitness vectors in HEA^T are very time consuming. This is mostly due to the system perspective models for visual and noise pollution, which calculate the amount of pollution generated in *each* ground square in *each* time interval, based on which UAVs that affect the ground square in a time interval. This is clearly a big task when there are many UAVs affecting ground squares in various parts of the environment.

Table 8.8: Runtimes of HEA^{C} for each test instance when modifying HEA parameters. Parameters with a "+" correspond to increased values compared to the base case (BC), while parameters with a "-" correspond to reduced values compared to the base case. The runtimes in each row below the "BC" row are given as multiples of the base case.

	Runtime		
Version	**TI1**	**TI2**	**TI3**
BC	185.8 hrs	132.3 hrs	113.4 hrs
N^{+}	1.518x	1.534x	1.551x
N^{-}	0.717x	0.695x	0.674x
p^{Mut+}	0.998x	0.970x	0.988x
p^{Mut-}	1.027x	1.015x	1.021x
No HEA^{F}	0.992x	0.982x	0.997x
Only mutating crossover	1.052x	1.036x	1.030x
Only regular crossover	0.983x	0.897x	0.967x

The apparent positive correlation between runtime and the number of affected ground squares is strengthened by the results in Tables 8.7 and 8.8. The most clear sign of this positive correlation, is the fact that changes in the ground square multiple λ results in the largest runtime effects in Table 8.7. Furthermore, Table 8.7 shows a clear tendency of longer runtimes when the visual and noise pollution thresholds decrease, i.e. R^{V-} and R^{N-}, and shorter runtimes when the thresholds increase. Decreases in the threshold values result in more ground squares being affected, while threshold increases result in less ground squares being affected. In contrast, the modified world and objective model parameters that do not influence the number of ground squares affected by the UAVs, i.e. α, β, δ and D^{Max}, have much smaller effects on the runtime. This is also the case for several of the modified algorithmic parameters, which runtimes are discussed in more detail in Section 8.6. The tables also show that the runtimes for TI2 are somewhat longer than those for TI3, despite the two test instances having many similarities. A natural reason for this is that since TI2 includes fictional no-flight zones, some of the UAVs in the system have to take longer paths, which thus results in more ground squares being affected and longer runtimes. Furthermore, we see that the runtimes for TI1 are longer than those for TI2 and TI3. This runtime difference is most likely due to the system of UAVs in TI1 having two additional passenger UAVs compared to TI2 and TI3, since passenger UAVs affect more ground squares than delivery UAVs. As described in Chapter 6, the number of ground squares affected by visual and noise pollution increases with the length and SPL of a UAV, respectively, and the passenger UAV type used in this thesis has greater length and SPL values than the delivery UAV type.

8.6 Importance of Components, Operators and Parameters of the Hybrid Evolutionary Algorithm

This section dives deeper into the inner workings of the HEA by investigating three aspects of the solution method. One aspect is how the inclusion of HEA^F runs in HEA^C affects the resulting system flight plans. As explained in Section 7.1, HEA^C calls HEA^F for each UAV in the system to generate an initial population. The HEA^F runs also need initial populations, and HEA^F uses ACO for the initial population generation. We have thus tested using the results from ACO to create the initial population for HEA^C directly. Another aspect investigated in this section is the effect of the chosen offspring generation operators, namely regular crossover and mutating crossover. As described in Section 7.1, regular crossover works by randomly combining segment paths for one UAV from the two selected parent individuals to create an offspring. Mutating crossover works by selecting a segment of one UAV's voyage and using ACO to alter the path taken on the segment, while all preceding segments use the segment paths from one parent individual and all succeeding segments use the segment paths from the other parent individual. To better understand the effects of the two crossover operators, HEA^C has been run using only one of the two operators for both the HEA^F and HEA^T parts of the algorithm. The last aspect investigated in this section is the effect of the modified HEA parameters, i.e. the population size of the HEA^T part and the mutation threshold for both the HEA^F and HEA^T parts.

Table 8.8 in Section 8.5 shows that modifying the population size parameter, N, of the HEA^T part of HEA^C appears to have a significant impact on runtime, while none of the other aspects considered in this section have a similarly sized effect. The table indicates a strong positive correlation between population size and runtime. Reducing the population size by a third also reduced the runtime by about a third, with multiples of 0.674-0.717 of the base cases. Similarly, increasing the population size by two thirds increased the runtime of the test instances by more than half, with runtime multiples of 1.518-1.551. These results are thus in line with the fitness computation being the most time consuming part of HEA^T, since each additional individual in the population results in an additional fitness computation.

For the crossover operators, Table 8.8 shows that using only regular crossover results in reduced runtimes for all test instances, with multiples of 0.897-0.983 of the base cases. Conversely, Table 8.8 shows that using only mutating crossover resulted in increased runtimes for all test instances, with multiples of 1.030-1.052 of the base cases. These runtime effects intuitively make sense, since using ACO to create a new path is more time consuming than combining existing segment paths. The runtime differences would most likely have been more significant if the UAVs had flown longer paths, as that would lead to ACO creating longer paths in each mutating crossover, which takes more time. The effect of modifying the mutation threshold p^{Mut} , i.e. how often mutating crossover is used, naturally has a similar, but weaker, runtime effect than only using one of the two crossover operators. Table 8.8 shows that reducing the mutation threshold, i.e. increasing the probability of creating offspring using mutating crossover, from the base value of 0.8 down to 0.5, increases the runtime with multiples between 1.015-1.027 of the base cases. Conversely, increasing the mutation threshold from 0.8 to 0.9 reduces the runtime with multiples between 0.970-0.998 of the base cases.

In regards to the effect of including the HEA$^{\mathrm{F}}$ component, Table 8.8 in Section 8.5 shows that the runtime reduction achieved by skipping HEA$^{\mathrm{F}}$ is minimal for all three test instances, with runtime multiples between 0.982-0.997 of the base cases. A likely reason for the small runtime effects is that computational results are stored during the runs of the HEA$^{\mathrm{F}}$, including during the initial population generation by the ACO. Preliminary testing showed that, similar as for HEA$^{\mathrm{C}}$, the objective computations constitute the most time-consuming part of the HEA$^{\mathrm{F}}$ runs. By storing the calculated values, the energy consumption and single-UAV perspective objective values associated with each vertex and edge in the graph are only calculated once for each type of UAV. Since the ACO runs are the first step in HEA$^{\mathrm{F}}$, the remaining parts of HEA$^{\mathrm{F}}$ can just look up the values that have already been calculated by the ACO run. This in turn makes the following steps of HEA$^{\mathrm{F}}$ fast. Since the system perspective calculations depend on where each of the UAVs in the system is located at each point in time, which differs between system flight plan individuals in the population, the same type of intermediate value storage and lookup is not possible for the HEA$^{\mathrm{T}}$ part of the HEA$^{\mathrm{C}}$. The system perspective fitness vector of each individual is stored in the HEA$^{\mathrm{T}}$ part however, and does hence not need to be recalculated if an individual stays in the population for multiple iterations.

While only the population size parameter seems to have a significant runtime effect, several of the aspects impact the solution quality. Figure 8.8 shows the objective values for the five system flight plans returned from HEA$^{\mathrm{C}}$ for the various versions of the test instances. Most notably, the runs using only mutation crossover appear to perform best overall, in terms of finding the lowest minimum values for the various objectives. It is even more important than the population size, which has a relatively small impact. The runs using only mutating crossover are best for all objectives for TI1 and TI2, and best for the risk and noise pollution objectives for TI3, with the reduced mutation threshold runs performing best on the other two objectives for TI3. Thus, it appears that finding the optimal ratio between regular and mutating crossover is the most important algorithmic choice. This result makes sense, since mutating crossover includes randomness in the ACO generated paths, thus introducing new paths which can help to avoid local optima. Conversely, Figure 8.8 shows that the runs using only regular crossover generally perform rather poorly, which makes sense since it only switches around existing

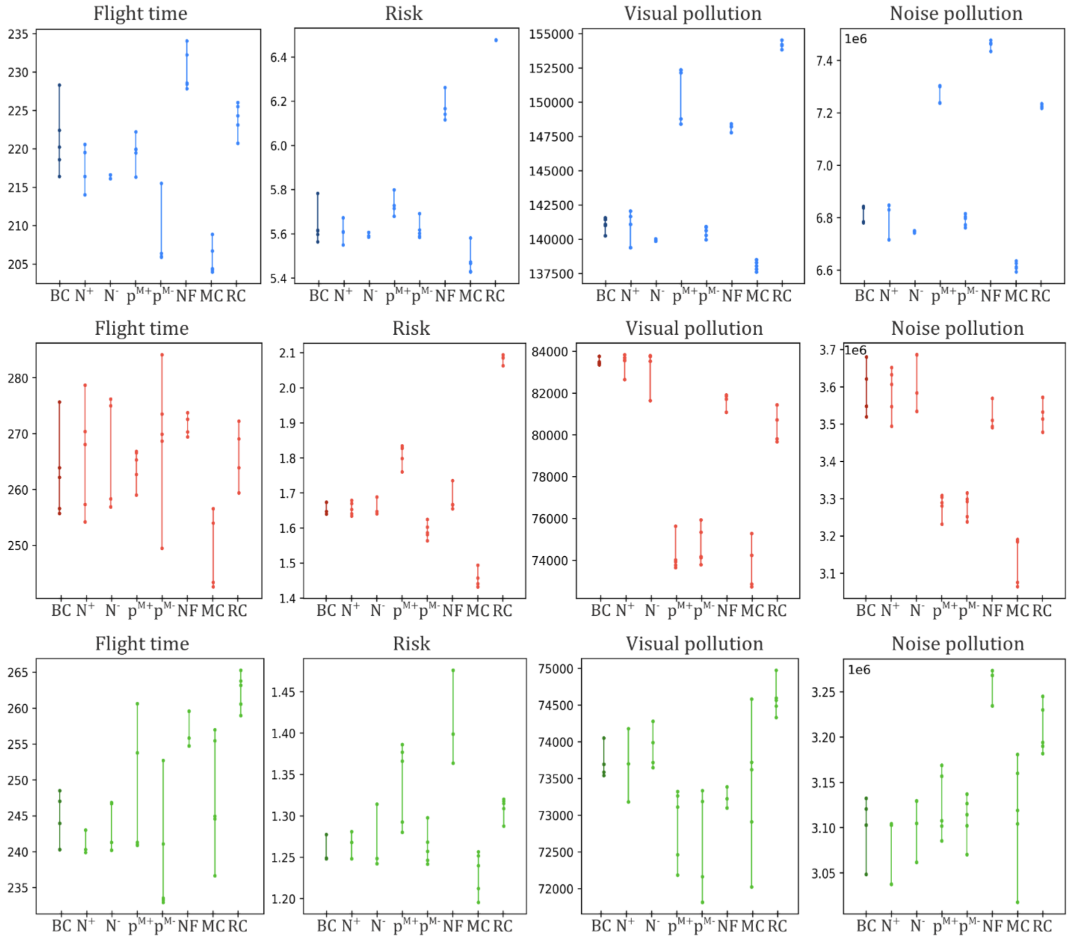


Figure 8.8: Illustration of the objective values of each of the five system flight plans returned from the HEA^C for the base case (BC) and various HEA^C versions with differing algorithmic parameter values. The objective values of TI1, TI2 and TI3 are shown in the blue, red and green plots, respectively. Parameters with a "+" correspond to increased values compared to the base case, while parameters with a "-" correspond to reduced values. Note that p^{Mut} has been shortened to p^M in the figure. Furthermore, we have NF: no HEA^F, MC: only mutating crossover and RC: only regular crossover. The "1e6" in the corner of the noise pollution plots denotes that the values are in millions. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

segment paths, which increases the chance of getting stuck in local optima. The results furthermore seem to indicate that the optimal ratio between the two crossover types is more inclined towards mutating crossover than the equal split tested by the reduced mutation threshold runs, i.e. it might be beneficial to set the mutation threshold lower than the 0.5 value tested as the reduced mutation threshold value in this thesis. Overall, the results hence indicate that it is possible to keep the population size, and thus the runtime, down while keeping the solution quality high by using mutating crossover often enough in the HEA^{C}.

It might not be beneficial to lower the mutation threshold down to 0 however, as indicated by the reduced threshold run having lower minimum values than the only mutating crossover run for flight time and visual pollution objectives for TI3. As previously mentioned, the runtime increase from using the mutating crossover operator more often will likely be more significant as the path lengths for the UAVs increase. Furthermore, since the mutating crossover operator only updates one randomly selected segment of one randomly selected UAV's voyage when generating a new offspring and there is randomness involved in the path generation by ACO, the propagation of improved segments for a given UAV throughout various system flight plans can be slow. To see this, assume the use of the mutating crossover operator leads to significant improvement in the path on segment s for UAV u for a new offspring individual p. It will take many offspring generations to significantly improve the population of system flight plans, since the good segment update on s must be recreated by the ACO for all system flight plans that do not have the individual p as a parent. It is furthermore not guaranteed that the mutating crossover operator will create such an improving path for a segment of a UAV's voyage for each offspring generation. Both the increased runtime and the potentially slow propagation of good paths may be mitigated by also using regular crossover. While the mutating crossover operator can ensure that local optima are avoided by introducing new segment paths for UAVs, the regular crossover can spread such new segment paths more quickly throughout the population and thus keep the runtime more manageable. Hence, the use of a combination of the two operators in the HEA might still be the preferred choice.

In regards to the effect of including HEA^{F} in HEA^{C}, Figure 8.8 indicates that the base case runs find lower minimum values for the risk and flight time objectives. The results are less clear for visual and noise pollution, however. TI2 has worse base case values for noise pollution compared to the no-HEA^{F} values, and both TI2 and TI3 have worse base case values for visual pollution. Thus, whether to include the HEA^{F} runs seems to result in a trade-off between the objectives. This trade-off can potentially be avoided by combining the results from running only ACO and running HEA^{F} when creating the initial population for HEA^{C}. This combination could consist of creating some initial individuals using only the ACO results and some using only the HEA^{F} results, or by combining paths from both ACO and HEA^{F} in the same individual. Such a combination of results from ACO and HEA^{F} in the initial population generation could thus potentially allow for the exploitation of the good qualities of both. Since TI1 has better values for the base case for all objectives in Figure 8.8, the inconsistencies in visual and noise pollution objectives for TI2 and TI3 could also be related to the solutions found by the HEA^{C} not being entirely stable. This possibility is discussed further in Section 8.7.

8.7 Stability of the Hybrid Evolutionary Algorithm

The randomness included in the HEA makes it important to assess the stability of the algorithm. To explore the stability of the HEA, we ran HEA^C ten times for the base case for TI1, without the seeding of random numbers described in Section 8.5. We first look at the runtime differences between the runs, and then the variation in objective values.

The runtimes of the HEA^C runs are displayed in Table 8.9. The maximum runtime variation is approximately 9% from the average runtime, thus indicating noticeable variations. A possible explanation for the variations is that some of the runs have found longer paths for the UAVs, resulting in more affected ground squares to calculate system perspective objective values for.

Table 8.9: Runtimes of HEA^C for different TI1 base case runs. TI1$_1$ (BC) is the run used as the base case in the other sections of the computational study.

Run	Runtime [hrs]	Runtime multiple
TI1$_1$ (BC)	185.8	1.000x
TI1$_2$	190.9	1.028x
TI1$_3$	183.1	0.985x
TI1$_4$	186.6	1.004x
TI1$_5$	165.3	0.890x
TI1$_6$	162.7	0.875x
TI1$_7$	166.1	0.894x
TI1$_8$	181.8	0.978x
TI1$_9$	169.1	0.910x
TI1$_{10}$	162.9	0.877x
Average [-/+ dev]	**175.4 [12.8/15.5]**	**0.944 [0.069/0.083]**

Figure 8.9 shows the objective values for each objective for the five system flight plans returned from each TI1 run. Similar as for the runtime, there is also variation in solution quality. While many of the runs have found similar minimum values for flight time, there is considerably more variation in the other three objectives. It may thus be beneficial to run the HEA^C multiple times. The orange lines in Figure 8.9 shows the objective values resulting from selecting the final five system flight plans from the set of all system flight plans returned from the ten TI1 runs. This combination-based approach is clearly able to extract the good qualities of the various TI1 runs, and results in the lowest minimum values for all objectives in Figure 8.9. It hence seems that the best performance for HEA^C can be achieved by using a meta-algorithm that runs the HEA^C multiple times, combines the result and selects the final set of system flight plans using the final selection method used by each HEA^C run, described in Section 7.1.

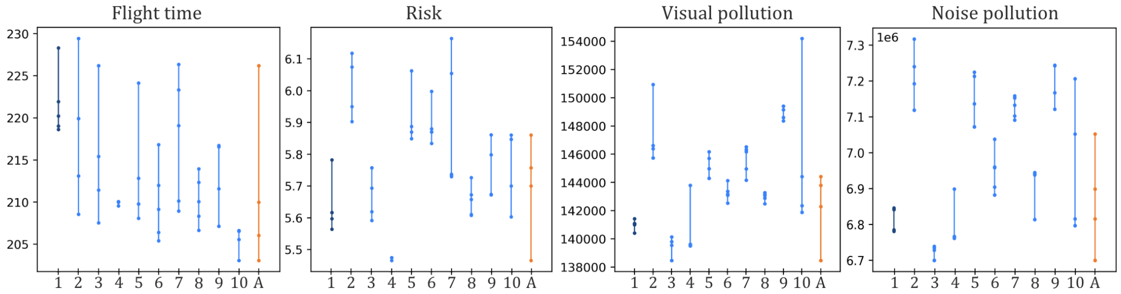


Figure 8.9: Illustration of the objective values for each of the five system flight plans returned from the HEA^C for ten base case runs of TI1. The "TI1$_x$" naming has been simplified to just "x" in the figure. "1" corresponds to the base case run used in the seeded version of the HEA^C. The orange line at the right end of each plot named "A" is the result of selecting five system flight plans from the set of all system flight plans returned from the ten runs of TI1. The "1e6" in the corner of the noise pollution plot denotes that the values are in millions. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

8.8 The Ground Square Size Trade-off

The number of ground squares affected by a system of UAVs has a significant impact on runtime, as explained in Section 8.5. The parameter that influences the ground squares the most is the ground square multiple λ, which decides the size of a ground square and thus also the total number of ground squares in the environment. Table 8.7 shows that increasing λ from the base value of 10 to 12, resulted in runtime reductions with multiples between 0.541-0.642 of the base case runs, while a reduction of λ down to 6 resulted in runtime increases with multiples between 1.990-2.093. Given the extensive runtimes of the HEA^C with $\lambda = 10$, a reduction to 6 seems unreasonable given the doubling in runtime. The speed-up from increasing λ to 12 is more interesting, however. Generally, a smaller value of λ results in more accurate computations of the objective values as explained in Subsection 8.1.1. It is therefore essential to find a good balance between runtime and computational accuracy, and an increase in λ from 10 to 12 requires further analysis to understand which value is more suitable.

We begin the analysis by studying the objective values. Since the size of a ground square directly affects the objective values, objective values are not directly comparable across HEA^C runs with different values for λ. To allow for such comparisons, the objective values for the system flight plans returned from the HEA^C runs with $\lambda = 10$ and $\lambda = 12$ have been recomputed using the set of ground squares with $\lambda = 6$. Figure 8.10 shows these comparable objective values for the three test instances for $\lambda \in \{6, 10, 12\}$. The HEA^C runs with $\lambda = 6$ are included for reference. For all the plots in Figure 8.10, the minimum objective values of the base case runs with $\lambda = 10$ are closer or equally close to the minimum objective values for the runs with $\lambda = 6$ compared to the runs with $\lambda = 12$. Thus, $\lambda = 10$ appears to be more suitable.

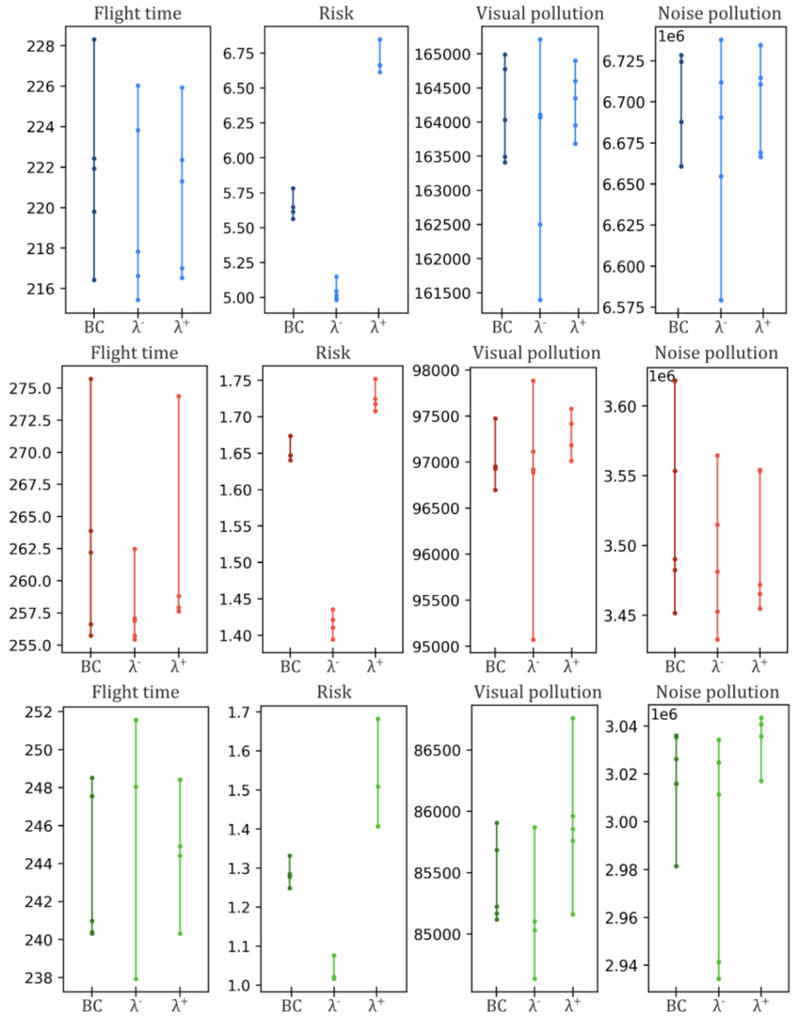
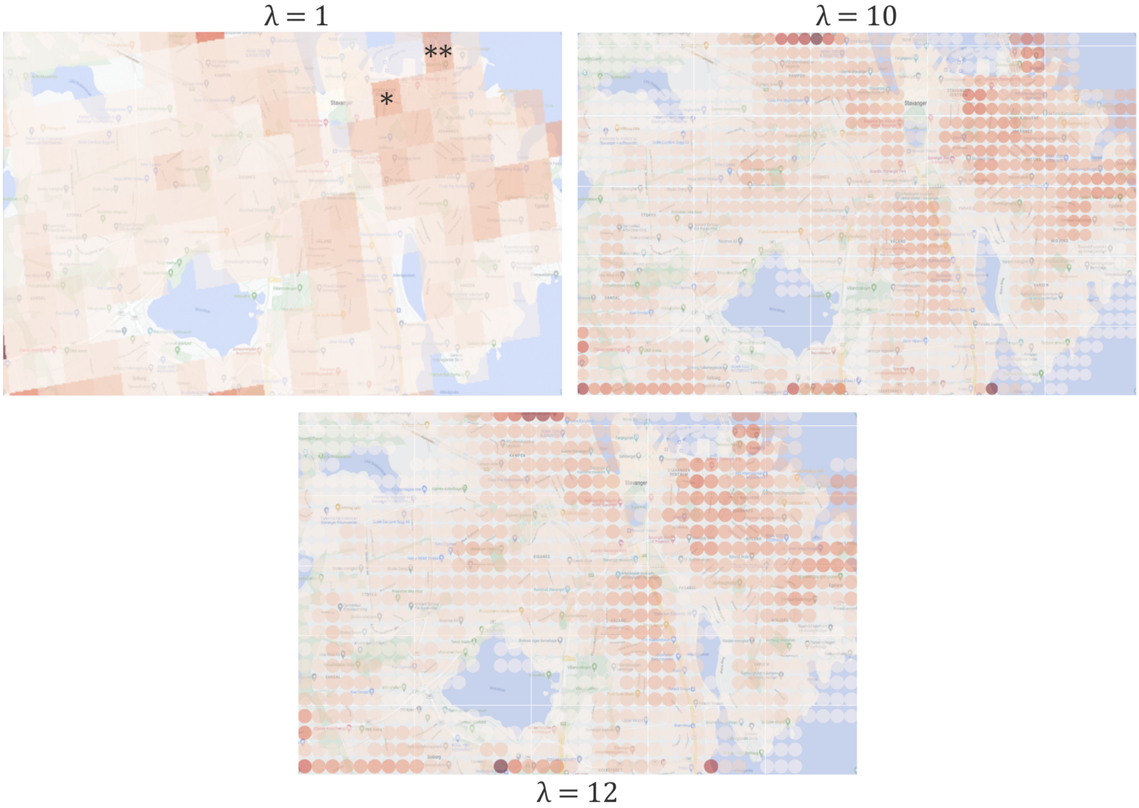


Figure 8.10: Illustration of the objective values for each of the five system flight plans returned from the HEA$^{\text{C}}$ for the base case (BC), for HEA$^{\text{C}}$ with λ reduced to 6 (λ^-) and for HEA$^{\text{C}}$ with λ increased to 12 (λ^+). The objective values for TI1, TI2 and TI3 are shown in the blue, red and green plots, respectively. The "1e6" in the corner of the noise pollution plots denotes that the values are in millions. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

To further investigate the differences between $\lambda = 10$ and $\lambda = 12$, we consider how the ground square size affects environment characteristics, i.e. population density and sheltering factor. Since these characteristics are associated with each ground square, increasing the size of a ground square means more information must be aggregated or combined for each ground square, possibly at the expense of accuracy. Figures 8.11 and 8.12 show the population density and sheltering factor, respectively, for λ values of 1, 10 and 12. Comparing the population densities and sheltering factors in Figures 8.11 and 8.12 for $\lambda = 1$ and $\lambda = 10$, the overall characteristics of the environment seems to be preserved when $\lambda = 10$. When further increasing λ to 12, we see many similarities with the figures for $\lambda = 10$, but also some differences.



$\lambda = 1$

$\lambda = 10$

$\lambda = 12$

Figure 8.11: Illustration of the population density in the urban environment with λ values of 1, 10 and 12. A darker color indicates a higher population density. The stars point to areas where the parts of the figure with $\lambda = 10$ and $\lambda = 12$ have clearly visible differences.

In Figure 8.11, the highly populated area marked with one star is still present when $\lambda = 10$, but seems to disappear when $\lambda = 12$. Similarly, the area marked with two stars is clearly visible when $\lambda = 10$, but has faded significantly more when $\lambda = 12$, meaning the population has become more evenly dispersed due to the size of the ground squares. In Figure 8.12, the area marked with stars all indicate areas where the plot with $\lambda = 12$ overestimates the sheltering factor compared to $\lambda = 10$. Overestimating the sheltering factor can be problematic, as it can make areas of the environment seem more safe to fly in, in terms of keeping objective values low, while in reality resulting in higher risk, visual pollution and noise pollution. This problem is clearly visible in Figure 8.10, with $\lambda = 12$ having significantly higher minimum values than $\lambda = 10$ for some of the objectives. Thus, $\lambda = 10$ seems to preserve environment characteristics to a larger degree than $\lambda = 12$. Overall, $\lambda = 10$ hence appears the most suitable, as it gives a satisfactory trade-off between runtime and accuracy.

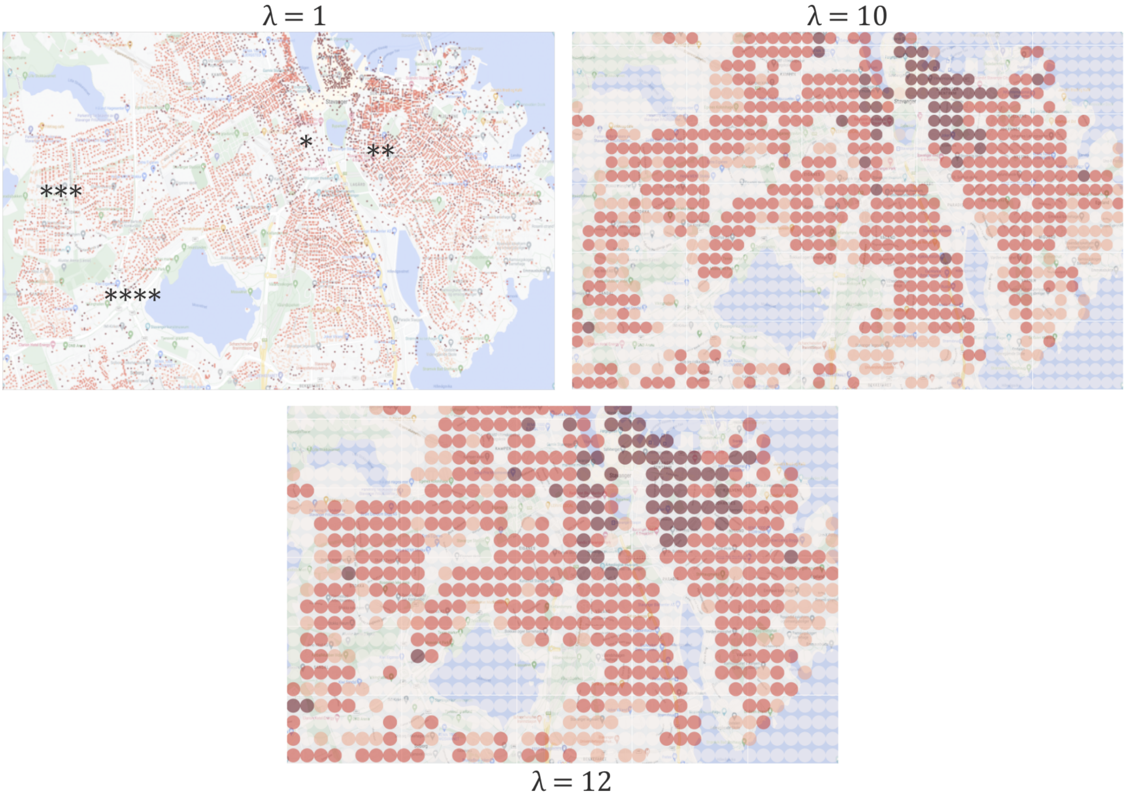


Figure 8.12: Illustration of the sheltering factor in the urban environment with λ values of 1, 10 and 12. A darker color indicates a higher sheltering factor. Given the large number of areas with the lowest possible sheltering factor when $\lambda = 1$, these areas are left without a color in the plot to make the remaining sheltering factors more visible. The stars point to areas where the parts of the figure with $\lambda = 10$ and $\lambda = 12$ have clearly visible differences.

8.9 System Behavior

The previous sections indicate that the HEA overall has the potential to be a suitable solution method for the static, multi-UAV path planning problem. This section investigates the system flight plans returned for the three test instances to evaluate the hypotheses stated in Section 8.2. Given the short flight distances between the locations for each UAV in the test instances, the energy constraint is never binding and is thus not discussed further. Note that since the system perspective objectives are calculated for a system flight plan as a whole, it can be difficult to pinpoint why a system flight plan does well for a given objective. Some UAVs' paths that give low values for an objective might be offset by other UAVs' paths giving a higher value for the same objective. It should also be noted that some of the system flight plans contain squiggles that are not necessarily advantageous in terms of the objectives, but originate from the ACO used as foundation when generating system flight plans. Furthermore, the path plots in this section show the final paths after all UAVs have reached their final locations. The start times of the UAVs have been set to allow simultaneous flight of UAVs in close proximity however, as explained in Section 8.2. References to figures when discussing UAV interactions hence concern UAV movements at the same points in time, if not clearly stated otherwise. Finally, this section only concerns HEA^C, and thus uses the term HEA. The two following subsections consider the hypotheses H1-H4 and H5-H8, respectively.

8.9.1 Interplay Between Objectives and Unmanned Aerial Vehicles

This subsection takes a closer look at the results from the base case runs of HEA to discuss hypotheses H1-H4. First we present some overarching results, followed by investigations of each of the four hypotheses.

Figure 8.13 shows the objective values for the five system flight plans returned for the base case of each test instance. Overall, we see that the HEA is able to find diverse trade-offs between the objectives, thus allowing a decision maker to decide which system flight plan best suits their preferences. While the value ranges for some of the objectives appear small in absolute values, we consider the ranges satisfactory given the short distances covered by the UAVs. The flight time objective constitutes the most simple and easily interpretable example of this. The passenger and delivery UAV types need 0.36 and 0.6 seconds to traverse a straight edge, respectively. Thus, the smallest and largest value ranges for flight time across the three test instances are equivalent to adding approximately 23-56 additional passenger UAV edges or 14-33 additional delivery UAV edges. This introduces path length increases of between 230-560 meters and 140-330 meters since the length of a straight edge is 10 meters, as described in Appendix D. These possible path length increases are considerable given the overall distances covered, shown in Figures 8.2 and 8.3.

We draw three further insights from Figure 8.13. The first additional insight is that system flight plans with a low risk value, often have higher values for visual and noise pollution. Such a negative correlation is likely related to the fact that while flying in close proximity to other UAVs can decrease both the visual and noise pollution objective values, flying closer than D^{Max} increases the risk objective values. Looking at the yellow points for TI2, it is also possible to have low values for risk, visual pollution and noise pollution simultaneously. A possible explanation for this is that the UAVs fly close enough to get significant reductions in visual and noise pollution due to simultaneous impact, but not close enough to be within each other's collision risk areas. Flying in close proximity to other UAVs is not always beneficial for the noise pollution objective however, which is related to the second additional insight from

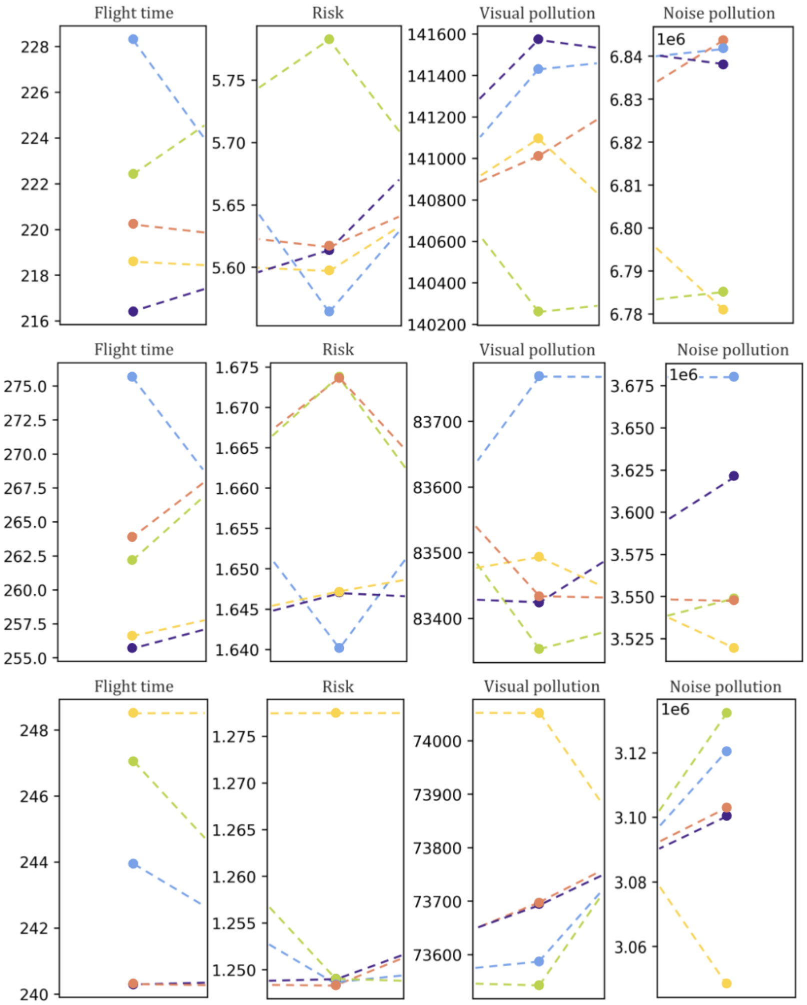


Figure 8.13: Illustration of the objective values for the five system flight plans for the base case of each test instance. Objective values for TI1, TI2 and TI3 are shown in the top, middle and bottom row, respectively. The color of a point within a row indicates the system flight plan it belongs to. The “1e6” in the corner of the noise pollution plots denotes that the values are in millions. The flight time unit is seconds, the risk unit is expected fatalities and the units for visual and noise pollution have no straight-forward interpretation. Note that the numeric values for risk are inflated due to the conservative estimation method and thus overestimates the actual number of fatalities.

the figure, namely that the objective values for visual and noise pollution can both be positively and negatively correlated. Since the threshold for visual pollution is considered for each UAV in isolation, the objective value is always reduced when UAVs fly in close enough proximity to affect the same ground squares within the same time intervals. This is not always the case for noise pollution, however. It is possible that two UAVs individually generate noise that is below the noise pollution threshold in a given ground square, but end up exceeding the threshold when they fly within close enough proximity. Flying in close proximity can hence result in the

UAVs generating noise pollution in more ground squares than when they are further apart. Thus, it is only beneficial that the UAVs fly in close proximity when both UAVs already generate noise exceeding the noise pollution threshold in a given ground square. Visual and noise pollution can hence be both positively and negatively correlated. The third additional insight from Figure 8.13 is that the flight time objective is not clearly correlated with the other objectives. The figure shows that both long and short flight times can result in high and low values for the other objectives. This seems reasonable since there exists various ways for the UAVs to fly that result in low values for the objectives. For risk, a short flight time is beneficial if the environment below the UAV is fairly homogeneous and there are no UAVs within the given UAV's collision risk area. In contrast, a longer flight time might for example be beneficial if it makes the UAV avoid the increased risk area of other UAVs. For visual and noise pollution, flying at high altitudes above ground, which is more time consuming, and flying closer to other UAVs, which can be less time consuming since it can be done at low altitudes, generally result in reduced objective values.

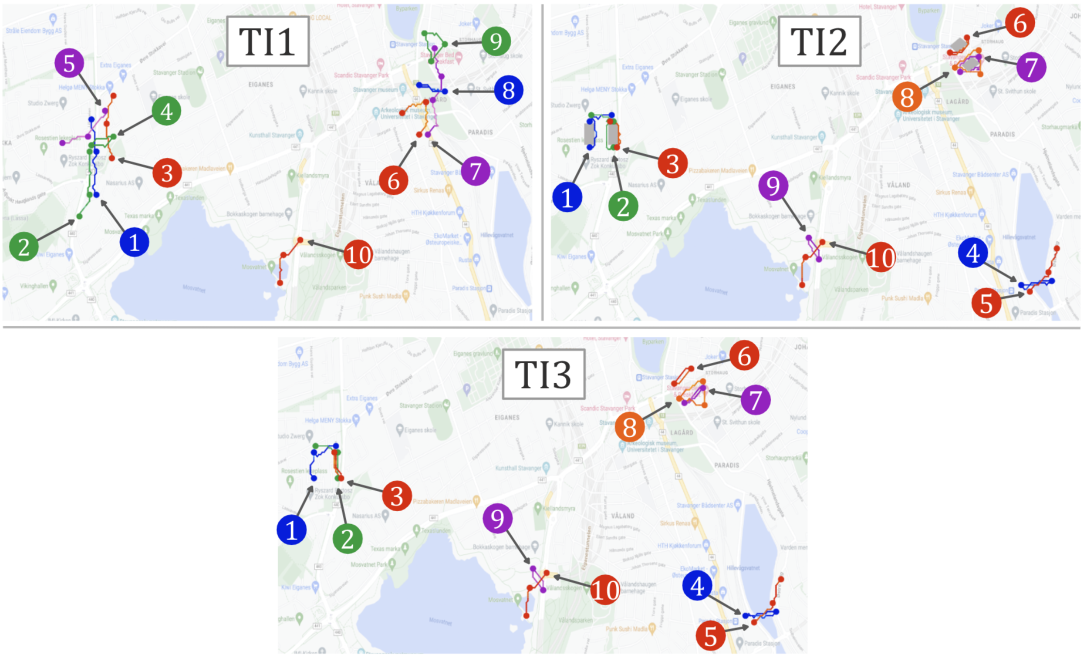


Figure 8.14: Illustration of a two dimensional path plot for one system flight plan returned from the HEA for each test instance. The "1.x"/"2.x"/"3.x" formulation of UAV names has been omitted for simplicity, and each number points at the corresponding start location for each UAV. Red, orange and purple dots and lines represent delivery UAVs, while blue and green dots and lines represent passenger UAVs. A lighter version of the colors correspond to the UAVs flying at a higher altitude above ground.

Having analyzed the objective values for the system flight plans for each test instance, we now take a closer look at the paths taken by each UAV. Figure 8.14 shows two dimensional path plots for one system flight plan for each test instance. We only show one system flight plan for each test instance, due to the system flight plans generally having many similarities. The differences between the system flight plans are discussed later. Some general insights can be gathered from the two dimensional path plots in Figure 8.14, however. The plots for TI2 and TI3 in the figure show that the UAVs in TI3 fly further apart when there are no fictional no-flight zones, which is also the case for the other system flight plans for the two test instances. This can be related to the short distances traveled by each UAV. With shorter distances, the increase in objective values from an increase in the path length become relatively larger than if the path was longer, thus making system flight plans with short path lengths more common. Another possible reason for the proximity differences between TI2 and TI3 can be that there exists an area outside two UAVs' collision risk areas where any relative positions of the two UAVs lead to the common affected ground squares staying the same. Since the system perspective objective models use the Euclidean distance from a UAV to the center of a ground square, there is a balance between being as far away from a ground square while simultaneously ensuring that the two UAVs affect the same ground squares. Thus, the path differences between the UAVs flying near fictional no-flight zones in TI2 compared to the same UAVs in TI3, can be related to the size of the area where two UAVs affect the same ground squares being larger than the distance between the pairs of fictional no-flight zones located in the same area.

To properly discuss hypotheses H1-H4, the paths taken by various UAVs must be studied in more detail. We thus take a closer look at the various clusters. Despite it being necessary to consider all UAVs in a system when discussing objective values, as stated in the beginning of this section, it makes sense to zoom in to clusters when investigating how the overflown area or the possibility of collisions affect UAV behavior.

Effect of Population Density and Sheltering Factor

To discuss hypothesis H1, i.e. that population density trumps the sheltering factor, we take a closer look at UAVs 1.6 and 1.9 in TI1. Both UAVs start in areas with high sheltering and population density and move to areas with lower values for both factors. Figure 8.15 shows a close up of the different path alternatives in the cluster containing UAVs 1.6-1.9. The figure indicates that both UAV 1.6 and 1.9 generally tries to fly out of the highly populated and sheltered area as fast as possible, keeping a low altitude and moving in a fairly straight line away from the area. Thus, the paths of the two UAVs in the various system flight plans indicates that hypothesis H1 holds. This result seems reasonable, as flying above low populated areas results in lower values for the risk, visual pollution and noise pollution objectives, while a high sheltering factor only reduces risk and visual pollution while having no impact on noise pollution. However, the results could also be related to the fact that the short path length overall leads to a relatively large impact on the objective values when the path length increases.

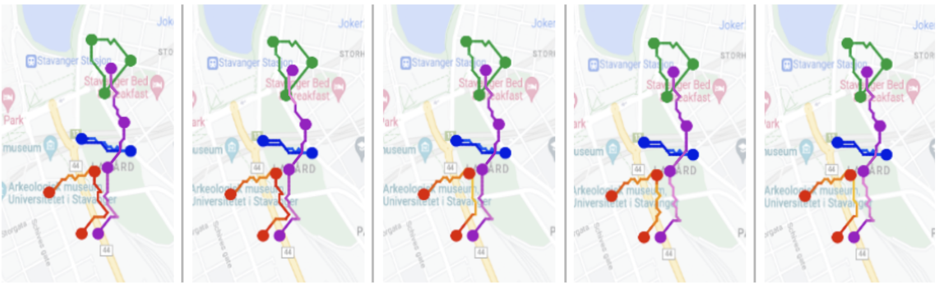


Figure 8.15: Illustration of two dimensional path plots for the cluster consisting of UAVs 1.6-1.9 from TI1. The red lines and dots represent UAV 1.6, purple lines and dots represent UAV 1.7, blue lines and dots represent UAV 1.8 and green lines and dots represent UAV 1.9. A lighter version of the respective colors correspond to the UAVs flying at a higher altitude above ground.

Effect of Proximity Differences

Hypothesis H2 states that a UAV flying far away from other UAVs in a system behaves differently when another UAV is introduced in its vicinity. Figure 8.16 shows the resulting paths for UAV 1.10 in TI1 compared to UAVs 2.10 and 3.10 in TI2 and TI3, respectively. For simplicity, the figure only shows a representative selection of system flight plans that covers the types of paths found in the various test instances. Parts a)-c) of the figure appear to indicate that UAV 1.10 flies more varied paths, many which include higher-altitude paths. In contrast, UAVs 2.10 and 3.10 have fairly homogeneous paths, all generally at low altitudes, as shown by parts d)-f) in the figure. Figure 8.2 thus seems to strengthen hypothesis H2, as the path choices are different for UAV 1.10 compared to UAVs 2.10 and 3.10.

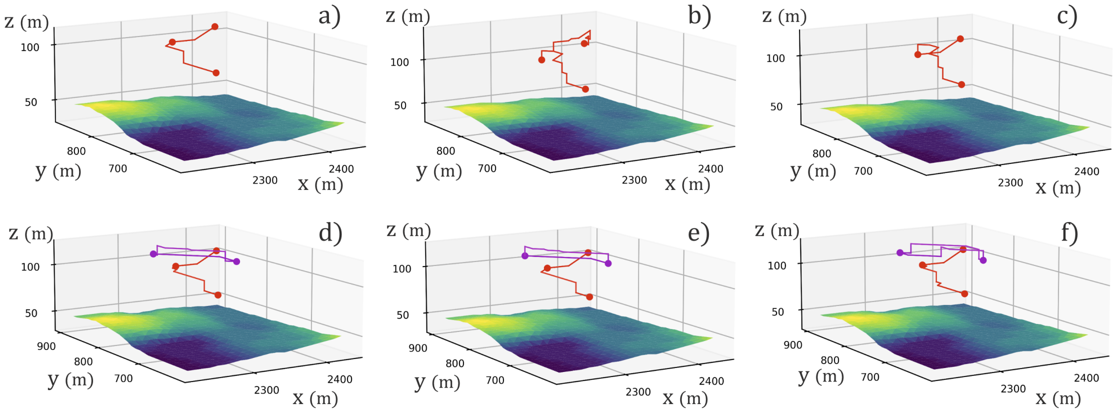


Figure 8.16: Illustration of three dimensional path plots for corresponding clusters for the test instances. Parts a)-c) show the path variations flown by UAV 1.10 in red, while parts d)-f) show the path variations flown by UAV 2.10 and 3.10 in red. The purple lines and dots in parts d)-f) represent UAVs 2.9 and 3.9. Note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

Effect of Importance of Efficiency on Collision Avoidance

Various UAVs in the three test instances can be studied to evaluate hypothesis H3, i.e. that UAVs with a lower importance of efficiency are the ones responsible for collision avoidance when encountering UAVs with higher importance of efficiency. We begin by looking at the UAVs in TI1. Unfortunately, the timing of UAVs 1.1 and 1.4 do not match up and they are hence never on collision course. Thus, we cannot use the interaction between these two UAVs to discuss H3. The high importance of efficiency-valued UAV 1.4 crosses paths with the low importance of efficiency-valued UAV 1.3 at the same point in time, however. Figure 8.17 shows the four different paths that UAV 1.3 takes in the five system flight plans for TI1, with the last system flight plan being omitted as UAV 1.3 flies the same full path as in the system flight plan in part a) of the figure. Hypothesis H3 is upheld by the two UAVs, i.e. the UAV with the lower importance of efficiency, shown in red in the figure, is responsible for avoiding collision with the UAV with the higher importance of efficiency, shown in green.

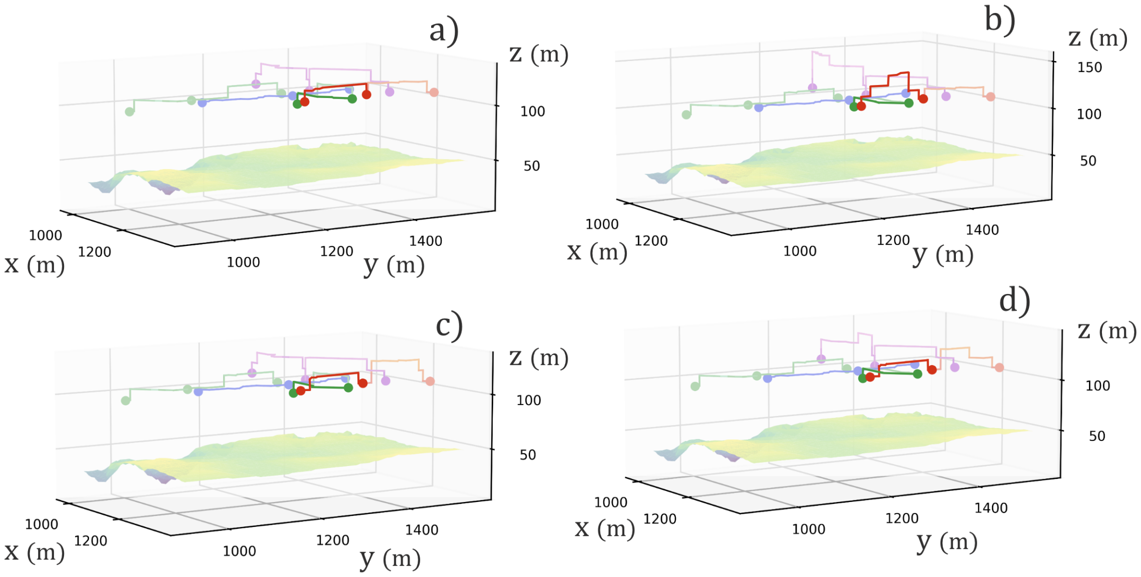


Figure 8.17: Illustration of three dimensional path plots for the cluster consisting of UAVs 1.1-1.5 from TI1. The red lines and dots represent UAV 1.3 and green lines and dots represent UAV 1.4. The rest of the UAVs, as well as the paths on other segments for UAVs 1.3 and 1.4, have been faded out for increased readability. Note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

The same result is seen for UAVs 2.4 and 2.5 in TI2 and UAVs 3.4 and 3.5 in TI3, shown in Figure 8.18. Two system flight plans have been omitted for TI2 since UAV 2.5 flies the same full path as in part a) for the omitted system flight plans, and two system flight plans have been omitted for TI3 since UAV 3.4 flies the same full path as in part c) for the omitted system flight plans. In parts a), b) and c) of the figure, UAV 2.5, which has a low importance of efficiency and paths shown in red, is responsible for collision avoidance with UAV 2.4, which has a high importance of efficiency and paths shown in blue. Conversely, when the importance of efficiency is switched around for the two UAVs in TI3, the opposite UAV is responsible for collision avoidance. For parts a) and b) of the figure, we can furthermore see that UAV 2.4 chooses to fly higher for a portion of its path on the segment after the collision segment. This might be due to the additional height reducing visual and noise pollution.

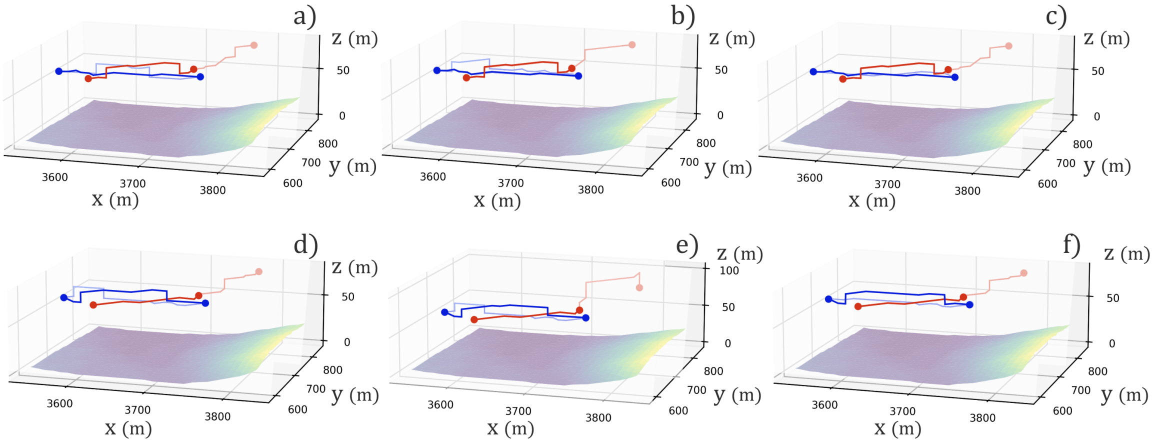


Figure 8.18: Illustration of three dimensional path plots for UAVs 2.4 and 2.5 from TI2 and UAVs 3.4 and 3.5 from TI3. The red lines and dots represent UAVs 2.5 in parts a)-c) and UAV 3.5 in parts d)-f). The blue lines and dots represent UAV 2.4 i parts a)-c) and UAV 3.4 in parts d)-f). The paths on other segments for the UAVs have been faded out for increased readability. Note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

One thing to note from the analysis of H3, is that each of the pairs of UAVs considered above contain one passenger and one delivery UAV. Since a passenger UAV generates higher increases in objective values with increased path length compared to a delivery UAV, and the low importance of efficiency-valued passenger UAV still performs the collision avoidance when a higher importance of efficiency-valued delivery UAV is in its way, the short distances traveled actually strengthen the hypothesis.

Effect of Unmanned Aerial Vehicle Types on Collision Avoidance

While hypothesis H3 appears to be strengthened by the results, we now consider if the same holds for H4, i.e given equal importance of efficiency values, delivery type UAVs are the ones responsible for collision avoidance when encountering passenger UAVs, due to the significant differences in UAV characteristics between the two UAV types. Figure 8.19 shows the four different paths that UAVs 1.1 and 1.5 take in the five system flight plans for TI1, with the last system flight plan being omitted as both UAVs fly the same full paths as in the system flight plan in part a) of the figure. The results in the figure appear to strengthen the hypothesis, since delivery UAV 1.5, shown in purple, is always responsible for the collision avoidance with passenger UAV 1.1, shown in blue.

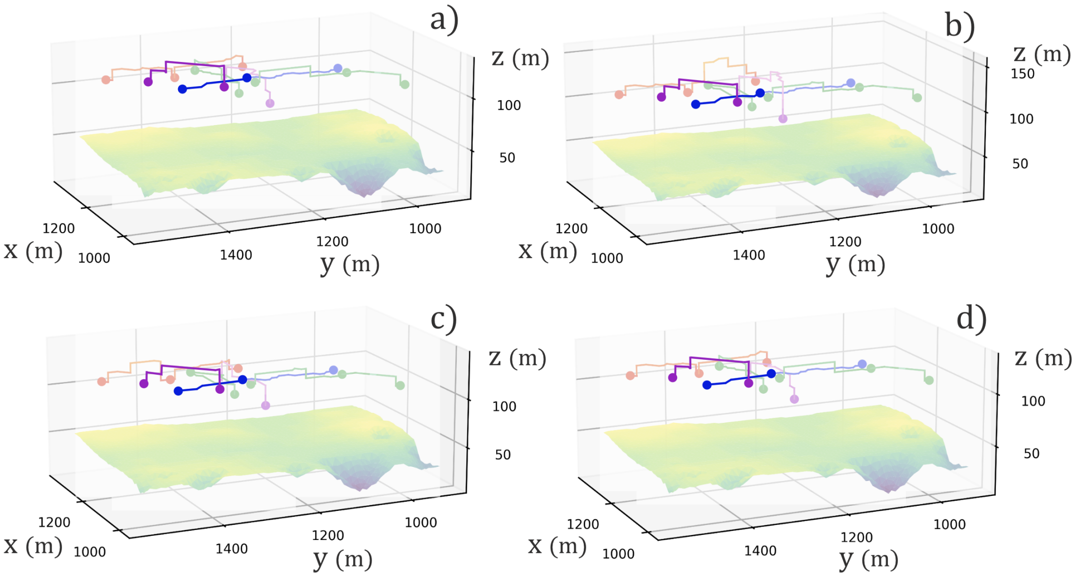


Figure 8.19: Illustration of three dimensional path plots for the cluster consisting of UAVs 1.1-1.5 from TI1. The purple lines and dots represent UAV 1.5 and blue lines and dots represent UAV 1.1. The rest of the UAVs, as well as the paths on other segments for UAVs 1.1 and 1.5, have been faded out for increased readability. Note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

While Figure 8.19 strengthens hypothesis H4, the results for delivery UAV 1.7 and passenger 1.8 are more unclear. The delivery UAV 1.7 flies upwards and passenger UAV 1.8 flies downwards when the two UAV cross paths in Figure 8.20, and thus it is not clear which UAV is responsible for the collision avoidance. At the same time, UAV 1.7, shown in purple in the figure, has to move to a higher altitude to reach its next location and it never flies higher than this given altitude. In contrast, UAV 1.8, shown in blue in the figure, appears to fly lower than necessary given the location it is flying to. This could thus indicate that it is in fact passenger UAV 1.8 that is responsible for collision avoidance, thus weakening hypothesis H4. It could also be the case that UAV 1.8 wants to fly at a low altitude above ground and is following the changes in the terrain, since the altitude above sea level of the ground becomes lower in the given area. Thus, the result in regards to H4 remains inconclusive. Similar to the other figures, one system flight plan has been omitted from Figure 8.20, since its full paths for UAVs 1.7, 1.8

and 1.9 are equal to part d) in the figure. In contrast to the comparison between delivery UAV 1.7 and passenger UAV 1.8, Figure 8.20 shows that delivery UAV 1.7 is always responsible for collision avoidance with passenger UAV 1.9 shown in green, in line with hypothesis H4.

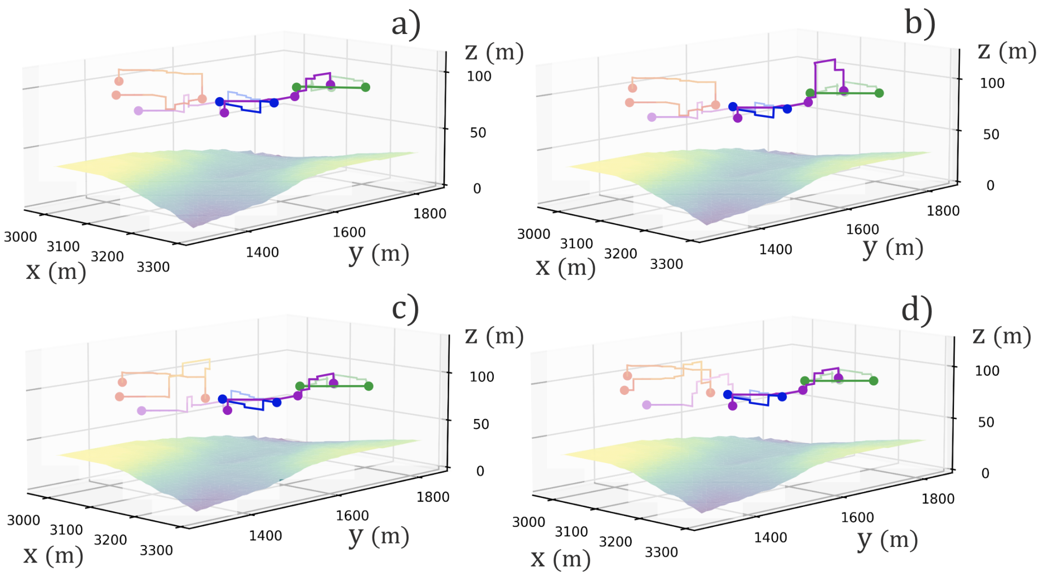


Figure 8.20: Illustration of three dimensional path plots for the cluster consisting of UAVs 1.6-1.9 from TI1. The purple, blue and green lines and dots represent UAV 1.7, 1.8 and 1.9, respectively. The rest of the UAVs, as well as the paths on other segments for UAVs 1.7-1.9, have been faded out for increased readability. Note that the ground in each part displays a more detailed version of the ground than the ground squares used for objective calculations. The color of the ground within each part of the figure ranges from dark blue to yellow, with a lighter color corresponding to a higher altitude above sea level.

8.9.2 Effect of Modifying World and Objective Model Parameters

This subsection investigates the effects of modifying the world and objective model parameters related to hypotheses H5-H8. Since H5-H8 regard entire systems of UAVs, we consider the behavior of all UAVs collectively when evaluating the hypotheses, as opposed to the interplay between a subset of UAVs in the system studied in H1-H4. A proximity measure is used to discuss each of these hypotheses, rather than path plots. Visual inspection of path plots is not a viable option to gain insight into overall changes in proximity due to the large number of UAVs within each test instance.

The proximity values are shown in Table 8.10. The proximity measure used to generate the values in the table finds the average proximity of the five system flight plans returned from a HEA run. For each system flight plan, the proximity is calculated by first aggregating the Euclidean distance in meters between each pair of UAVs within the same cluster every 0.2

seconds. The distance is only calculated for UAVs flying between locations in the air, i.e. the distance to a UAV performing a service is ignored for the duration of the service time and UAVs that have either not begun or have finished their voyages are also ignored at a given 0.2 second checkpoint. A cluster consisting of only one UAV has a proximity value of 0. After finding the proximity value of each cluster in a test instance, the cluster values are aggregated to obtain the proximity value of the system flight plan. The 0.2 second time interval length has been set to ensure that no UAV passes through more than one vertex between checkpoints, given the straight edge traversal times of 0.36 and 0.6 seconds for a passenger and a delivery UAV, respectively. A higher value for the proximity measure implies that the UAVs fly further apart, while a lower value implies that they fly closer to each other. The reason why this proximity measure was chosen, and not e.g. to aggregate the distances between every UAV and its closest UAV, is that the system perspective models calculate objective values for the system of UAVs as a whole. Thus we believe the chosen measure is more appropriate than a measure that looks at UAVs within the same clusters separately.

Table 8.10: Proximity values of the base case (BC) and various HEA versions with modified world and objective model parameter values. Parameters with a "+" correspond to increased values compared to the base case, while parameters with a "-" correspond to reduced values. For each test instance, both the proximity value and the deviation of the proximity value from the base case is shown.

	TI1		**TI2**		**TI3**	
Version	**Proximity [m]**	**Dev. [m]**	**Proximity [m]**	**Dev. [m]**	**Proximity [m]**	**Dev. [m]**
BC	467 550	0	246 804	0	242 224	0
α^+	452 602	-14 948	249 348	2 544	242 976	753
α^-	459 027	-8 523	251 695	4 891	242 373	149
β^+	458 987	-8 563	247 200	396	239 096	-3 128
β^-	459 719	-7 831	250 111	3 307	241 786	-438
δ^+	458 406	-9 144	250 969	4 165	241 312	-912
δ^-	463 567	-3 983	254 026	7 222	241 693	-531
D^{Max+}	470 269	2 719	248 317	1 513	241 639	-585
D^{Max-}	457 012	-10 538	248 013	1 209	245 006	2 782
R^{V+}	466 127	-1 423	249 864	3 060	245 118	2 894
R^{V-}	454 872	-12 678	245 845	-959	237 934	-4 290
R^{N+}	460 578	-6 972	247 673	869	240 397	-1 827
R^{N-}	459 919	-7 631	246 623	-180	241 045	-1 179

Overall, Table 8.10 shows that both increasing and decreasing the values of many of the modified parameters result in decreases in the proximity value, i.e. the UAVs fly closer than in the base case, for TI1 and TI3, while the parameters result in increases in the proximity value for TI2. This could indicate that there are other factors, such as environment or UAV characteristics, that impact the effect the modified parameters have on system flight plans and that these variations have not been sufficiently accounted for in the test instances. It could also be the case that the selected base values for the various parameters are not as suitable as we assumed. In the remainder of this subsection, we use two UAVs flying 10 meters apart as an equivalent to the proximity measure, denoted *tenquivalent*, to make the interpretation of the numerical values in Table 8.10 more intuitive. Two UAVs flying 10 meters apart for 1 second, results in a proximity value of $10 \cdot \frac{1}{0.2} = 50$. Thus, a proximity value of e.g. 5000 corresponds to $\frac{5000}{50} = 100$ seconds of two UAVs flying 10 meters apart, i.e. 100 seconds of tenquivalent.

Impact of Modifying α

Hypothesis H5 states that α, i.e. the factor for maximum reduction in mean time between falldowns due to the possibility of collisions, affects the proximity of UAVs. The α^+ and α^- rows in Table 8.10 show that α can affect the proximity in both directions. The table shows that the UAVs in TI1 fly somewhat closer for both modifications of α, indicated by a negative number in the deviation column. In contrast, both modifications of α for TI2 and TI3 result in positive deviation values, i.e. the UAVs fly further apart compared to the base case. The deviations in proximity across the test instances for both increasing and reducing α correspond to 3-299 seconds, i.e. up to 5 minutes, of tenquivalent. The deviations when looking at TI3 in isolation correspond to 3-15 seconds of tenquivalent, which weakens H5. However, when looking at all three test instances, it is clear that α can have a significant impact on proximity, thus strengthening H5.

Impact of Modifying β

Similar to the results for α, the discount factor for the system perspective visual pollution model has varying effects on proximity. The results in Table 8.10 for β^+ and β^- show that the UAVs in TI1 and TI3 fly in closer proximity when both increasing and reducing β, while the UAVs in TI2 fly further apart for both modifications of β. Thus, it appears as if changing β in isolation does not guarantee a given behavior among the UAVs in terms of flying closer or further apart. The deviations across the test instances for both reducing and increasing β correspond to 8-171 seconds, i.e. up to 2.9 minutes, of tenquivalent, showing that modifications of β can have both small and big impact on proximity. Hypothesis H6, i.e. that β affects proximity of UAVs, thus appears to hold.

Impact of Modifying δ

To evaluate H7, i.e. that the length of a time interval, δ, affects the proximity of UAVs, we again look at Table 8.10. Similar as for α and β, the results show that changes in δ have an effect on the proximity of UAVs, combined with variations across the test instances regarding whether the UAVs fly closer or further apart. It makes sense that changes in δ can have different effects on proximity as e.g. noise pollution values can both increase and decrease when δ increases, as explained in Subsection 8.1.1, and thus support both more and less proximity. Similarly, a given δ can lead to both underestimation and overestimation of the risk values. Since δ decides how often the relative positions of UAVs are checked, an increased δ leads to the same mean time between falldowns being used for a longer period of time. Thus, if the distances between UAVs are greater than the collision risk threshold at the two checkpoints at the start and end of a time interval, an increased δ leads to a lower risk value than the true value if the UAVs were in fact closer than the collision risk threshold somewhere between the two checkpoints. The deviations when changing δ correspond to 11-183 seconds, i.e. up to 3.0 minutes, of tenquivalent across the test instances. It thus seems that δ has the potential to impact proximity quite significantly, however it might not always do so.

Impact of Modifying Thresholds

The final hypothesis from Section 8.2, H8, is that the thresholds for collision risk, visual pollution and noise pollution, i.e. D^{Max}, R^V and R^N, affect the proximity of UAVs. Looking at Table 8.10, we see that the proximity values are lower when reducing both R^V and R^N, across all three test instances. It should however be noted that for TI2, there is only a small deviation in proximity from the base case when reducing R^N. A possible reason for the lower proximity values when the threshold values are lower, is that flying closer may result in an increase in the proportion of common ground squares among the UAVs' affected ground squares. When increasing R^V and R^N we see both higher and lower proximity values, however. Thus, the thresholds for visual and noise pollution can impact proximity in both directions. The deviations in proximity values across the test instances when changing R^V and R^N correspond to 19-254 seconds, i.e. up to 4.2 minutes, of tenquivalent and 4-153 seconds, i.e. up to 2.5 minutes, of tenquivalent, respectively. This thus indicates that changing these thresholds can have a significant impact on proximity, but that it is not always the case. For D^{Max} we also see an impact on proximity both towards flying closer and further apart, and there is no clear trend for the direct effects of D^{Max} on proximity. The deviations across the test instances when changing D^{Max} correspond to 12-211 seconds, i.e. up to 3.5 minutes, of tenquivalent, and D^{Max} can therefore greatly impact proximity. At the same time, D^{Max} does not always have a great impact on proximity, and the differences in proximity impact make sense when looking at the risk model in Section 6.3. If most UAVs fly at distances that are always within D^{Max-} or always outside D^{Max+} when modifying the parameter, it makes sense with small changes in proximity. If many UAVs fly at distances between the different D^{Max} values, however, it makes sense with larger changes in proximity. This could again indicate that the sensitivity to changes in D^{Max} are different for different intervals.

8.10 Discussion

The main focus of the computational study has been to investigate how the introduction of a system of UAVs affects UAV behavior, as explained in Section 8.2. This section first provides a summary of the findings in regards to the hypotheses, and then takes a broader look at the results in regards to widespread UAV usage.

8.10.1 The Hypotheses Reviewed

To get a more detailed picture of what impacts the system flight plans returned from the HEA^C, we developed a set of eight hypotheses. The first four hypotheses regard the system behavior for the base cases, while the last four consider the effects that different world and objective model parameters have on system behavior. Beginning with hypothesis H1, which states that population density trumps the sheltering factor, the hypothesis generally appears to be strengthened by the test instances. This indicates that environment characteristics surrounding a UAV are still considered despite other UAVs flying close by. It should be noted that the short distances traveled by the UAVs may have intensified the population versus sheltering results.

Hypothesis H2, i.e. that a UAV flying far away from other UAVs in a system behaves differently when another UAV is introduced in its vicinity, appears to be strengthened by the results from the test instances. The results indicate that flight patterns that are no longer beneficial due to the introduction of the additional UAV are discarded. Combined with the insights from hypothesis H1, it appears as if the system effects influence UAV behavior to a larger degree than environment characteristics, but environment characteristics still play an important role.

All the test instances indicate that hypothesis H3, which states that UAVs with a lower importance of efficiency are the ones responsible for collision avoidance when encountering UAVs with higher importance of efficiency, holds. The validation of H3 is not necessarily entirely positive however, as it may be an indication that the HEA^C is inclined to finding paths that do well for the flight time objective at the cost of the other objectives, similar to the insight for HEA^F in Section 8.3.

Hypothesis H4 states that given equal importance of efficiency values, UAV characteristics determine which UAV is responsible for collision avoidance when two UAVs meet. As previously mentioned, this hypothesis is equivalent to stating that delivery UAVs are responsible for collision avoidance when encountering passenger UAVs. H4 holds for two of the three pairs of UAVs testing this hypothesis. This could indicate that there are other factors that affect which UAV is responsible for collision avoidance to a larger degree than the increased values for the

UAV characteristics.

Hypotheses H5-H8 state that α, β, δ and the thresholds D^{Max}, R^V and R^N affect the proximity of UAVs. The proximity measure appears to strengthen all four hypotheses since all the modified parameters are able to significantly impact UAV proximity. The parameter values do not always have a great impact on proximity, however. Furthermore, there are no clear trends regarding whether the proximity increases or decreases when changing a parameter. As explained in Subsection 8.1.1, the values for the world and objective model parameters considered in H5-H8 will be decided by policy makers. It is thus crucial that policy makers understand the effects of altering the parameters values, to ensure the outcome is as intended. Thus, it appears as if further testing of the parameters considered by H5-H8 is necessary to gain more insights into their effects, potentially in more controlled and homogeneous environments, to reduce the possibility of other factors than the parameter in question affecting the results. It is also crucial to look at parameter values in conjunction. For example, both α and D^{Max} directly impact the risk objective. It is thus important to perform further testing on different combinations of these parameters, as well as other parameter combinations affecting different objectives.

Overall, the computational study has shown that the introduction of a system of UAVs has a great impact on UAV behavior. By accounting for collision avoidance and using system perspective equations to calculate objective values, the paths taken by each UAV can differ significantly from the paths taken if each UAV was considered in isolation. Thus, it is essential to take a system perspective when planning paths for a system of UAVs.

8.10.2 Beyond the Skies: The Bigger Picture

In addition to studying the system flight plans in and of themselves, it is also important to consider the effects of widespread UAV usage from a broader perspective. From an ethical point of view, allowing risk, visual pollution and noise pollution to take any value might not be desirable. As explained in Sections 2.6 and 2.7, visual and noise pollution can have negative health effects, and the same clearly applies for risk. Regulations for safety and allowable noise levels already exist for other means of transportation, and extending the model in this thesis with upper limits for risk, visual pollution and noise pollution could thus be beneficial. By studying what value levels for the objectives that result in adverse health effects, policy makers can introduce suitable limits that cannot be exceeded by the system flight plans. Using risk as an example, such an approach ensures a decision maker that regardless of which system flight plan is chosen, a statistical maximum number of expected fatalities will never be exceeded.

There is ongoing work and still a long way to go to develop effective UAV regulations, as described in Section 2.3. Researching future UAV regulations is thus essential from a political and legal viewpoint. The research should focus on understanding how various laws, regulations and limitations affect the behavior of UAVs, to ensure that the introduced policies have the desired effect. If behavioral effects are not taken into account and regulations are introduced without understanding the consequences, new regulations could possibly have no effect or the opposite of the desired effect. The results from hypotheses H5-H8 described above, which we consider to be important parameters for policy makers to understand, makes it clear that further assessment of the parameters is necessary.

Finally, it is worth making a note of the environmental dimension of UAV usage. This thesis focuses solely on the societal, i.e. human, impact of system flight plans, thus no attention is given to animals or plant life. While no-flight zones in nature conservation areas can help protect the animals in these areas, as mentioned in Section 6.3, animals living in other urban areas such as various types of birds, squirrels and pets are not protected nor accounted for in this thesis. As an example, consider an area with zero population density, such as a nature area consisting of a body of water or covered in trees and bushes, which is located in between more populated areas. If a UAV has to fly across these areas, it would naturally choose to fly above the nature area, as the zero population density is helpful given the incorporation of social impact in the objective models in this thesis. Thus, the plants and animals present in the area can be affected by the UAV. As the number of UAVs flying in urban areas increases, it is thus important to be aware of, and possibly account for, the environmental impact of UAVs, to avoid unnecessary disturbances or adverse effects on animal or plant life.

Chapter 9

Concluding Remarks

The use of UAVs for transportation in urban environments is expected to see significant growth in the future, thus also increasing the impact of UAV transport on the population in these areas. Path planning models that account for social impact are hence paramount, and this thesis proposes a model that finds socially acceptable solutions, called *system flight plans*, to the static, multi-UAV path planning (MUAVPP) problem in an urban environment. The model balances four objectives; flight time, risk, visual pollution and noise pollution, and accounts for the effect that simultaneous impact from multiple UAVs have on each objective. Furthermore, the energy capacity of each UAV is incorporated as constraints. A hybrid evolutionary algorithm (HEA) consisting of the non-dominated sorting genetic algorithm III (NSGA-III) and ant colony optimization (ACO) has been developed to solve the problem.

This thesis makes multiple contributions to existing literature. The literature review uncovered that visual and noise pollution have not been modeled as part of a path planning problem before, and the risk UAVs pose to humans has received little attention in UAV path planning literature. Allowing characteristics to vary along a voyage for a UAV also constitutes a contribution. This thesis allows the UAV payload weight and the importance of efficiency, i.e. the importance of short flight time, to vary along a voyage. Furthermore, to the best of our knowledge, using an evolutionary algorithm to solve a multi-objective, multi-vehicle problem in an urban environment is a novel proposition.

The HEA is applied to various test instances of the MUAVPP problem in a three dimensional urban environment. Overall, the solution method returns system flight plans with diverse trade-offs between the four objectives. The set of system flight plans are furthermore of a manageable size, thus facilitating a thorough and transparent decision making process. On a technical level, there are a couple of insights worth noting however. The HEA needs long runtimes even for relatively small test instances, in terms of distances traveled by each UAV in a system. The long runtime is mostly caused by the way simultaneous impact from multiple UAVs is accounted for in the objective models. The results also show that the solution method can be further improved. The amount of mutation taking place in the NSGA-III part of the HEA run should be increased and a meta-algorithm combining results from multiple runs of HEA on the same problem instance can improve both performance and stability.

The results from running HEA gave insights into a set of hypotheses, which overarchingly tests how the introduction of a system of UAVs affects UAV behavior. The first insight is that the environment characteristics affecting individual UAVs are still accounted for when a system of other UAVs are introduced. Conversely, a UAV flying by itself far away from other UAVs in a system, appears to fly significantly different paths when another UAV is introduced in its vicinity, thus indicating that the system effects might be considered more important than environment characteristics. An important aspect of studying systems of UAVs is the possibility of collisions. A further insight from the results is that the UAV that deviates from its path to avoid collision when two UAVs meet seem to depend on the importance of efficiency and UAV characteristics of the two UAVs. Low importance of efficiency-valued UAVs seem to be responsible for collision avoidance when encountering UAVs with a higher importance of efficiency, and smaller, lighter-weight and less noisy UAVs tend to be responsible for collision avoidance when encountering larger, heavier and more noisy UAVs. Furthermore, the evaluations of the hypotheses gave insights into the impact of using different values for world and objective model parameters. The values set for the maximum reduction in mean time between falldowns due to the possibility of collisions, the visual pollution discount factor, the length of a time interval and the thresholds for collision risk, visual pollution and noise pollution can all have a significant impact on the proximity of UAVs. That being said, the direct effects of varying each of the parameters are not clear from the evaluations, and further testing on the parameters should thus be performed.

In conclusion, the model proposed in this thesis appears to provide a socially conscious way to utilize the low-altitude urban airspace, by solving the static, multi-UAV path planning problem balancing flight time, risk, visual pollution and noise pollution criteria. The extensive runtimes is a clear weakness of the proposed solution method, and must hence be improved to facilitate real-world applicability. Be that as it may, this thesis still proves that it is possible to incorporate social acceptability when navigating urban skies.

Chapter 10

Future Research

The static, multi-UAV path planning problem considers a wide variety of aspects. Thus, there exist several avenues for future research to create more effective models for UAV path planning in an urban environment, both from a societal and from a scientific viewpoint.

Creating more realistic and advanced models for risk, visual pollution, noise pollution and energy consumption constitutes a possible area of future research. Both the visual and noise pollution objectives can be made time dependent, which further opens up the possibility of allowing UAVs to vary their flight speeds and wait in the air to avoid collisions. The quantification of visual and noise pollution can also be extended to differentiate the degree of pollution based on the UAV use case, since medical UAVs are considered more socially acceptable, as described in Section 2.6. The quantification of noise pollution can also be improved by e.g. accounting for the effect of varying UAV payloads or the existing noise levels in an area. The perceived noise from a UAV is likely to be lower in areas with high pre-existing noise levels, and could thus reduce the noise pollution. In regards to risk, further developments can be made for both air and ground risk. The probability of fatality for air risk is assumed to be constant in this thesis, but it is also possible to make it variable and dependent on e.g. the sheltering factor. The quantification of ground risk could be extended to consider various UAV descent models, as opposed to our assumption that the UAVs fall vertically straight down.

Another possible area of future research is to incorporate the interaction between the objectives, since e.g. the noise of a UAV can be perceived as worse if an individual is also able to see the UAV, as described in Section 2.6. It is also possible to differentiate the objective values for risk, visual pollution and noise pollution based on who is affected, i.e. assigning worse values if the same individual is affected twice compared to if two different individuals are affected once each. Furthermore, the system perspective equations proposed in Chapter 6 are complex, resulting in the computation of the system objective values being quite time-consuming. Thus, one area of future research is to test other models for system perspective quantifications of the objectives, i.e. of how the simultaneous impact from multiple UAVs affect risk, visual pollution and noise pollution. As discussed in Subsection 8.10.2, including upper limit constraints on the objectives or accounting for the environmental impact of UAV usage on nature life also constitute possible extensions to the model. In addition, the model should also be updated as new regulations are introduced to keep it up to date.

Future research can also expand on the environment. In reality, the UAVs will operate in a dynamic environment. Unforeseen events can take place during the execution of a system flight plan, e.g. the need for an emergency medical UAV to make a delivery to a person in need, or due to other aerial vehicles and birds that move around in the low-altitude airspace in urban environments. Furthermore, the problem can incorporate stochasticity, e.g. through variable edge flight times due to weather conditions or variable population in ground squares based on the time of day. Thus, studying the MUAVPP problem in a dynamic or stochastic environment, or both, can increase real world applicability. Further developments can also be made to the feasible moves of the UAVs, e.g. allowing for diagonal movement up and down in the graph in addition to horizontal diagonals. Such an extension will increase the complexity of the graph however, and must thus be weighted against the increased realism of the UAV movement.

Bibliography

Aasvang, G., Engdahl, B., & Krog, N. (2022). Støy, helseplager og hørselstap i Norge. Retrieved March 24, 2023, from https://www.fhi.no/nettpub/hin/miljo/stoy/

Abeywickrama, H. V., Jayawickrama, B. A., He, Y., & Dutkiewicz, E. (2018). Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access*, *6*, 58383–58394.

Ahmed, N., Islam, M. N., Tuba, A. S., Mahdy, M., & Sujauddin, M. (2019). Solving visual pollution with deep learning: A new nexus in environmental management. *Journal of environmental management*, *248*, 1–9.

AiRMOUR. (2022). Towards increased urban air mobility competence. Retrieved November 15, 2022, from https://airmour.eu/project-description/

Alexander, W. N., & Whelchel, J. (2019). Flyover noise of multi-rotor suas. *Inter-Noise and Noise-Con Congress and Conference Proceedings*, *259*(7), 2548–2558.

Alexopoulos, C., & Griffin, P. (1992). Path planning for a mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics*, *22*(2), 318–322.

Amazon. (2022). Amazon reveals the new design for prime air's delivery drone—here's your first look. Retrieved November 11, 2022, from https://www.aboutamazon.com/news/transportation/amazon-prime-air-delivery-drone-reveal-photos

Aviation Environment Federation. (n.d.). Understanding Aircraft Noise. Retrieved March 16, 2023, from https://www.aef.org.uk/guides/understanding-aircraft-noise/

Bai, H., Fan, T., Niu, Y., & Cui, Z. (2022). Multi-UAV Cooperative Trajectory Planning Based on Many-Objective Evolutionary Algorithm. *Complex System Modeling and Simulation*, *2*(2), 130–141. https://doi.org/10.23919/CSMS.2022.0006

Baker, R. (n.d.). Addition of sound intensities. Retrieved February 5, 2023, from https://personalpages.manchester.ac.uk/staff/richard.baker/BasicAcoustics/5_addition_of_sound_intensities.html

Balasingam, M. (2017). Drones in medicine—the rise of the machines. *The International Journal of Clinical Practice*, *71*.

Baloch, G., & Gzara, F. (2020). Strategic network design for parcel delivery with drones under competition. *Transportation Science*, *54*(1), 1–297.

Bellingham, J., Tillerson, M., Richards, A., & How, J. P. (2003). Multi-task allocation and path planning for cooperating UAVs. In *Cooperative control: Models, applications and algorithms* (pp. 23–41). Springer.

Bortoff, S. (2000). Path planning for UAVs. *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No.00CH36334)*, *1*(6), 364–368.

Burgess, M. (2016). DHL's delivery drone can make drops quicker than a car. Retrieved September 26, 2022, from https://www.wired.co.uk/article/dhl-drone-delivery-germany

Chen, J., Du, C., Zhang, Y., Han, P., & Wei, W. (2021). A clustering-based coverage path planning method for autonomous heterogeneous UAVs. *IEEE Transactions on Intelligent Transportation Systems*, 1–11.

Chen, P., Huang, Y., Papadimitriou, E., Mou, J., & van Gelder, P. (2020). Global path planning for autonomous ship: A hybrid approach of fast marching square and velocity obstacles methods. *Ocean Engineering*, *214*, 107793.

Chen, X., Xu, R., & Zhao, J. (2017). Multi-objective route planning for UAV. *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, 1023–1027.

Chen, Y., Han, J., & Zhao, X. (2012). Three-dimensional path planning for unmanned aerial vehicle based on linear programming. *Robotica*, *30*(5), 773–781.

Christian, A. W., & Cabell, R. (2017). Initial investigation into the psychoacoustic properties of small unmanned aerial system noise. *23rd AIAA/CEAS aeroacoustics conference*, 4051.

Cluff, J. (2021). New: A giant drone that carries one passenger. Retrieved September 26, 2022, from https://explorersweb.com/jetson-one-giant-passenger-drone/

Colorni, A., Dorigo, M., Maniezzo, V., et al. (1991). Distributed optimization by ant colonies. *Proceedings of the first European conference on artificial life*, *142*, 134–142.

Crumley, B. (2022). AiRMOUR tests drones in medical emergency supplies, personnel transport. Retrieved September 30, 2022, from https://dronedj.com/2022/09/27/aimour-drones-medical-emergencies/

Crumley, B. (2021). Swedish drone trial rushes defibrillators to heart attack victims faster. Retrieved September 30, 2022, from https://dronedj.com/2021/08/27/swedish-drone-trial-rushes-defibrillators-to-heart-attack-victims-faster/

Dalamagkidis, K., Valavanis, K. P., & Piegl, L. A. (2008). Evaluating the risk of unmanned aircraft ground impacts. *16th Mediterranean Conference on Control and Automation*.

Dasdemir, E., Köksalan, M., & Tezcaner Öztürk, D. (2020). A flexible reference point-based multi-objective evolutionary algorithm: An application to the UAV route planning problem. *Computers Operations Research*, *114*, 104811.

Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2000). A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii. In M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo & H.-P. Schwefel (Eds.), *Parallel problem solving from nature ppsn vi* (pp. 849–858). Springer Berlin Heidelberg.

Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, *18*(4), 577–601. https://doi.org/10.1109/TEVC.2013.2281535

Demir, E., Bektaş, T., & Laporte, G. (2014). A review of recent research on green road freight transportation. *European Journal of Operational Research*, *237*(3), 775–793.

Di Franco, C., & Buttazzo, G. (2015). Energy-aware coverage path planning of UAVs. *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 111–117.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269–271.

Dijkstra, L., & Poelman, H. (2014). A harmonised definition of cities and rural areas: The new degree of urbanisation. Retrieved February 2, 2022, from https://www.researchgate.net/publication/263488028_A_harmonised_definition_of_cities_and_rural_areas_the_new_degree_of_urbanisation

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on evolutionary computation*, *1*(1), 53–66.

EASA. (2013). *Acceptable means of compliance for large aeroplanes cs-25 amendment 13* (tech. rep.). European Union Aviation Safety Agency.

EASA. (2015). *Concept of operations for drones - a risk based approach to regulation of unmanned aircraft* (tech. rep.). European Union Aviation Safety Agency.

EASA. (2022). Easa publishes first guidelines on noise level measurements for drones below 600kg. Retrieved February 25, 2023, from https://www.easa.europa.eu/en/newsroom-and-events/press-releases/easa-publishes-first-guidelines-noise-level-measurements-drones

EASA. (2021). *Urban air mobility survey evaluation report* (tech. rep.). European Union Aviation Safety Agency.

EHang. (2016). EHang AAV: The era of urban air mobility is coming. Retrieved November 11, 2022, from https://www.ehang.com/ehangaav/

European Commission. (2021a). Commission implementing regulation (eu) 2021/664 of 22 april 2021 on a regulatory framework for the u-space. Retrieved March 7, 2023, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0664&from=EN

European Commission. (2021b). Commission implementing regulation (eu) 2021/665 of 22 april 2021 amending implementing regulation (eu) 2017/373 as regards requirements for providers of air traffic management/air navigation services and other air traffic management network functions in the u-space airspace designated in controlled airspace. Retrieved March 7, 2023, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0665&from=EN

European Commission. (2021c). Commission implementing regulation (eu) 2021/666 of 22 april 2021 amending regulation (eu) no 923/2012 as regards requirements for manned aviation operating in u-space airspace. Retrieved March 7, 2023, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32021R0666&from=EN

European Commission. (2022). A drone strategy 2.0 for a smart and sustainable unmanned aircraft eco-system in europe. Retrieved February 6, 2023, from https://ec.europa.eu/info/law/better-regulation/have-your-say/initiatives/13046-A-Drone-strategy-20-for-Europe-to-foster-sustainable-and-smart-mobility_en

European Commission. (2019a). The rules and procedures for the operation of unmanned aircraft, no. 2019/947. Retrieved March 7, 2023, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R0947&from=EN

European Commission. (2008). *Scientific committee on emerging and newly identified health risks* (tech. rep.). European Commission.

European Commission. (2019b). Unmanned aircraft systems and on third-country operators of unmanned aircraft systems. Retrieved March 7, 2023, from https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32019R0945&from=EN

Everdrone. (n.d.). Everdrone - our service. Retrieved October 25, 2022, from https://everdrone.com/our-service/

Feist, J. (2022). Drone taxi service – passenger drones. Retrieved September 26, 2022, from https://www.dronerush.com/drone-taxi-passenger-drones-10666/

FHI. (2021). *Kartlegging av kostholdsvaner og kroppsvekt hos voksne i Norge basert på selvrapportering* (tech. rep.). Folkehelseinstituttet.

Fügenschuh, A., & Müllenstedt, D. (2015). Flight planning for unmanned aerial vehicles. *IMA Conference on Mathematics in Defence 2015*.

Gasparetto, A., Boscariol, P., Lanzutti, A., & Vidoni, R. (2015). Path planning and trajectory planning algorithms: A general overview. In *Motion and operation planning of robotic systems: Background and practical approaches* (pp. 3–27). Springer International Publishing.

GoogleMaps. (2023). Google maps. Retrieved March 9, 2023, from https://www.google.com/maps

Grandview Research. (2020). *Commercial drone market size, share trends analysis report by product (fixed-wing, rotary blade, hybrid), by application, by end-use, by region, and segment forecasts, 2021 - 2028* (tech. rep.). Grandview Research.

Hasak, L., & Toomarian, E. Y. (2022). Brains are bad at big numbers, making it impossible to grasp what a million covid-19 deaths really mean. Retrieved February 7, 2023, from https://theconversation.com/brains-are-bad-at-big-numbers-making-it-impossible-to-grasp-what-a-million-covid-19-deaths-really-means-179081

Hayat, S., Yanmaz, E., Brown, T. X., & Bettstetter, C. (2017). Multi-objective UAV path planning for search and rescue. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 5569–5574.

Helse- og omsorgsdepartementet, Justis- og beredskapsdepartementet, Klima- og miljødepartementet, Kommunal- og moderniseringsdepartementet, Nærings- og fiskeridepartementet & Samferdselsdepartementet. (2018). Norges dronestrategi. Retrieved February 25, 2023, from https://www.regjeringen.no

Hentschel, M., Lecking, D., & Wagner, B. (2007). Deterministic path planning and navigation for an autonomous fork lift truck. *IFAC Proceedings Volumes*, *40*(15), 102–107.

Heutschi, K., Ott, B., Nussbaumer, T., & Wellig, P. (2020). Synthesis of real world drone signals based on lab recordings. *Acta Acustica*, *4*(6), 24.

Hu, J., & Lanzon, A. (2018). An innovative tri-rotor drone and associated distributed aerial drone swarm control. *Robotics and Autonomous Systems*, *103*, 162–174.

Hu, L., Naeem, W., Rajabally, E., Watson, G., Mills, T., Bhuiyan, Z., & Salter, I. (2017). Colregs-compliant path planning for autonomous surface vehicles: A multiobjective optimization approach. *IFAC-PapersOnLine*, *50*(1), 13662–13667.

Jaimes, A. L., Martınez, S. Z., Coello, C. A. C., et al. (2009). An introduction to multiobjective optimization techniques. *Optimization in Polymer Processing*, 29–57.

Jensen, C. U., Panduro, T. E., & Lundhede, T. H. (2014). The vindication of don quixote: The impact of noise and visual pollution from wind turbines. *Land economics*, *90*(4), 668–682.

Jhanwar, D. (2016). Noise pollution: A review. *Journal of Environment Pollution and Human Health*, *4*(3), 72–77.

Jung, S., & Kim, H. (2017). Analysis of Amazon prime air UAV delivery service. Retrieved November 11, 2022, from https://www.researchgate.net/publication/317389269_Analysis_of_Amazon_Prime_Air_UAV_Delivery_Service

Kjær, B. (n.d.). Sound power and sound pressure explained. Retrieved November 11, 2022, from https://www.bksv.com/en/knowledge/blog/sound/sound-power-sound-pressure

Klima- og miljødepartementet. (2021). *Retningslinje for behandling av støy i arealplanlegging (t-1442/2021)* (tech. rep.). Klima- og miljødepartementet.

Konopka, B. (2022). Is it a bird? Is it a plane? No, it's a Jetson ONE! Europe's first 'passenger drone' takes to the skies. Retrieved September 26, 2022, from https://www.thefirstnews.com/article/is-it-a-bird-is-it-a-plane-no-its-a-jetson-one-europes-first-passenger-drone-takes-to-the-skies-27156

Kraus, J., Kleczatský, A., & Hulínská, Š. (2020). Social, technological, and systemic issues of spreading the use of drones. *Transportation Research Procedia*, *51*, 3–10.

Lamont, G. B., Slear, J. N., & Melendez, K. (2007). UAV swarm mission planning and routing using multi-objective evolutionary algorithms. *2007 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making*, 10–20.

Li, P., Huang, X., & Wang, M. (2011). A novel hybrid method for mobile robot path planning in unknown dynamic environment based on hybrid dsm model grid map. *Journal of Experimental & Theoretical Artificial Intelligence*, *23*(1), 5–22.

Lin, C. E., & Shao, P.-C. (2020). Failure analysis for an unmanned aerial vehicle using safe path planning. *Aerospace Research Central*.

Liu, W., Li, S., Zhao, F., & Zheng, A. (2009). An ant colony optimization algorithm for the multiple traveling salesmen problem. *2009 4th IEEE Conference on Industrial Electronics and Applications*, 1533–1537.

Long, M. (2014). 2 - Fundamentals of Acoustics. In *Architectural acoustics (second edition)* (pp. 39–79). Academic Press.

Lutkevich, B. (2021). Drone (UAV). Retrieved September 29, 2022, from https://www.techtarget.com/iotagenda/definition/drone

Ma, Y., Hu, M., & Yan, X. (2018). Multi-objective path planning for unmanned surface vehicle with currents effects. *ISA Transactions*, *75*, 137–156.

Majeed, S., Sohail1, A., Qureshi, K. N., Kumar, A., Iqbal, S., & Lloret, J. (2020). Unmanned aerial vehicles optimal airtime estimation for energy aware deployment in IoT-enabled fifth generation cellular networks. *EURASIP Journal on Wireless Communications and Networking*.

Maristany de las Casas, P., Sedeño-Noda, A., & Borndörfer, R. (2021). An improved multiobjective shortest path algorithm. *Computers Operations Research*, *135*, 105424.

Meier, P. (2015). UAVs and humanitarian response. In *Drones and aerial observation: New technologies for property rights, human rights, and global development. a primer* (pp. 57–61). New America.

Merriam-Webster. (n.d.). Drone. Retrieved September 29, 2022, from https://www.merriam-webster.com/dictionary/drone

Modares, J., Ghanei, F., Mastronarde, N., & Dantu, K. (2017). Ub-anc planner: Energy efficient coverage path planning with multiple drones. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 6182–6189.

Mohamed, M. A. S., Ibrahim, A. O., Dodo, Y. A., & Bashir, F. M. (2021). Visual pollution manifestations negative impacts on the people of Saudi Arabia. *International Journal of Advanced and Applied Sciences*, *8*(9), 94–101.

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.

Norsk forening mot støy. (2018). Droner gir mye støyplager pr. desibel. Retrieved November 5, 2022, from https://www.stoyforeningen.no/Nyheter/Droner-gir-mye-stoeyplager-pr.-desibel

Ntakolia, C., & Lyridis, D. V. (2022). A comparative study on ant colony optimization algorithm approaches for solving multi-objective path planning problems in case of unmanned surface vehicles. *Ocean Engineering*, *255*, 111418. https://doi.org/https://doi.org/10.1016/j.oceaneng.2022.111418

OpenStreetMap. (n.d.). OpenStreetMap. Retrieved October 31, 2022, from https://www.openstreetmap.org/relation/10150658#map=10/59.0995/5.8152

Peng, C., Huang, X., Wu, Y., & Kang, J. (2022). Constrained multi-objective optimization for UAV-enabled mobile edge computing: Offloading optimization and path planning. *IEEE Wireless Communications Letters*, *11*(4), 861–865.

Portella, A. A. (2007). Evaluating commercial signs in historic streetscapes: The effects of the control of advertising and signage on user's sense of environmental quality. *Doctoral thesis, Oxford Brookes University*.

Posea, P. (2021). The complete history of drones (1898 - 2021). Retrieved September 26, 2022, from https://dronesgator.com/the-history-of-drones/

Primatesta, S., Rizzo, A., & la Cour-Harbo, A. (2020). Ground risk map for unmanned aircraft in urban environments. *Journal of Intelligent & Robotic Systems*, *97*(3), 489–509.

Qadir, Z., Zafar, M. H., Moosavi, S. K. R., Le, K. N., & Mahmud, M. A. P. (2022). Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment. *IEEE Internet of Things Journal*, *9*(14), 12505–12514.

Ren, Q., Yao, Y., Yang, G., & Zhou, X. (2019). Multi-objective path planning for UAV in the urban environment based on cdnsga-ii. *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 350–3505.

Roca-Riu, M., & Menendez, M. (2019). Logistic deliveries with drones: State of the art of practice and research. *19th Swiss Transport Research Conference (STRC 2019)*.

Rosser, J. C., Vignesh, V., Terwilliger, B. A., & Parker, B. C. (2018). Surgical and medical applications of drones: A comprehensive review. *Robotics and Autonomous Systems*, *22*.

Rutledge, K., McDaniel, M., Teng, S., Hall, H., Ramroop, T., Sprout, E., Hunt, J., Boudreau, D., & Costa, H. (2022). Urban area. Retrieved September 27, 2022, from https://education.nationalgeographic.org/resource/urban-area

Safe To Fly. (n.d.). Safe to fly map. Retrieved November 1, 2022, from https://www.safetofly.no/

Salford Acoustics. (n.d.). Decibel scale. Retrieved November 18, 2022, from https://salfordacoustics.co.uk/sound-waves/waves-transverse-introduction/decibel-scale

Samferdelsdepartementet. (2015). Forskrift om luftfartøy som ikke har fører om bord mv. Retrieved March 7, 2023, from https://lovdata.no/dokument/SF/forskrift/2015-11-30-1404

Samferdelsdepartementet. (1996). Lov om luftfart (luftfartsloven). Retrieved March 7, 2023, from https://lovdata.no/dokument/NL/lov/1993-06-11-101/*

Schäffer, B., Pieren, R., Hayek, U. W., Biver, N., & Grêt-Regamey, A. (2019). Influence of visibility of wind farms on noise annoyance–a laboratory experiment with audio-visual simulations. *Landscape and urban planning*, *186*, 67–78.

Schäffer, B., Pieren, R., Heutschi, K., Wunderli, J. M., & Becker, S. (2021). Drone noise emission characteristics and noise effects on humans—a systematic review. *International Journal of Environmental Research and Public Health*, *18*(11), 5940.

Schierbeck, S., Hollenberg, J., Nord, A., Svensson, L., Nordberg, P., Ringh, M., Forsberg, S., Lundgren, P., Axelsson, C., & Claesson, A. (2021). Automated external defibrillators delivered by drones to patients with suspected out-of-hospital cardiac arrest. *European Heart Journal*, *43*(15), 1478–1487.

Sedov, L., Polishchuk, V., Maury, T., Ulloa, M., & Lykova, D. (2021). Qualitative and quantitative risk assessment of urban airspace operations. *SESAR Innovation Days (SID 2021)*.

Sedov, L., Polishchuk, V., & Vishwanath, B. (2021). Ground risk vs. efficiency in urban drone operations. *Fourteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2021)*.

Shao, S., Peng, Y., He, C., & Du, Y. (2020). Efficient path planning for UAV formation via comprehensively improved particle swarm optimization. *ISA Transactions*, *97*, 415–430.

Shivgan, R., & Dong, Z. (2020). Energy-efficient drone coverage path planning using genetic algorithm. *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*, 1–6.

Sims, J. (2022). Are flying taxis the future of transport? experts say that passenger drones may take off as early as 2040 – making for a us$1.5 trillion market. Retrieved September 26, 2022, from https://www.scmp.com/magazines/style/news-trends/article/3176015/are-flying-taxis-future-transport-experts-say-passenger

Singh, I. (2021a). $4-per-mile flying taxis to tackle Los Angeles, Miami traffic? Retrieved September 26, 2022, from https://dronedj.com/2021/06/14/flying-taxis-los-angeles-miami/

Singh, I. (2021b). EHang is developing a new 2-seater drone taxi for inter-city travel. Retrieved September 26, 2022, from https://dronedj.com/2021/05/26/ehang-vt-30-drone-taxi/

Skenteris, K., Mirasgedis, S., & Tourkolias, C. (2019). Implementing hedonic pricing models for valuing the visual impact of wind farms in greece. *Economic Analysis and Policy*, *64*, 248–258.

Soltani, A., & Fernando, T. (2004). A fuzzy based multi-objective path planning of construction sites. *Automation in Construction*, *13*(6), 717–734.

Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N., & Feyzeau, P. (2013). Path planning: A 2013 survey. *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*, 1–8.

Srinivas, N., & Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, *2*(3), 221–248. https://doi.org/10.1162/evco.1994.2.3.221

Statistisk Sentralbyrå. (2022). Kart SSB. Retrieved November 4, 2022, from https://kart.ssb.no/

Statistisk Sentralbyrå. (2012). Statistisk årbok 2013, tabell 109: Egenrapportert høyde, vekt og svømmeferdighet for sesjonspliktige, etter fylke. 2012. Retrieved November 11, 2022, from https://www.ssb.no/a/aarbok/tab/tab-109.html

Statistisk Sentralbyrå. (2021). Tettsteders befolkning og areal. Retrieved November 25, 2022, from https://www.ssb.no/befolkning/folketall/statistikk/tettsteders-befolkning-og-areal

Stavanger kommune. (n.d.). AiRMOUR. Retrieved September 26, 2022, from https://www.stavanger.kommune.no/en/samfunnsutvikling/stavanger-smart-city/smart-city-projects/airmour/

Sumartono, S. (2009). Visual pollution in the contex of conflicting design requirements. *Jurnal Dimensi Seni Rupa dan Desain*, *6*(2), 157–172.

Susini, A. (2015). A technocritical review of drones crash risk probabilistic consequences and its societal acceptance. *Lnis*, *7*, 27–38.

Tezcaner, D., & Köksalan, M. (2011). An interactive algorithm for multi-objective route planning. *Journal of optimization theory and applications*, *150*(2), 379–394.

Thomas, K. (2022). Quantifying visual pollution from urban air mobility. *Bachelor thesis, Linköping University, Universidad Politécnica de Madrid*.

Toor, A. (2016). This startup is using drones to deliver medicine in Rwanda. Retrieved September 26, 2022, from https://www.theverge.com/2016/4/5/11367274/zipline-drone-delivery-rwanda-medicine-blood

Topographic Map. (n.d.). Topographic map. https://en-gb.topographic-map.com/

Torija, A. J., & Clark, C. (2021). A psychoacoustic approach to building knowledge about human response to noise of unmanned aerial vehicles. *International Journal of Environmental Research and Public Health*, *18*(2), 682.

Torija, A. J., Self, R. H., & Lawrence, J. L. (2019). Psychoacoustic characterisation of a small fixed-pitch quadcopter. *Inter-Noise and Noise-Con Congress and Conference Proceedings*, *259*(8), 1884–1894.

Transport Up. (2019). The EHang 216 takes flight in Vienna. Retrieved November 11, 2022, from https://transportup.com/headlines-breaking-news/the-ehang-216-takes-flight-in-vienna/

Treichel, J., & Körper, S. (2019). Untersuchung der geräuschemission von drohnen. *Lärmbekämpfung*, *14*(4).

Vance, A. (2021). Flying taxi startup Whisper Aero aims to keep noise levels down. Retrieved September 20, 2022, from https://www.bloomberg.com/news/articles/2021-07-12/flying-taxi-startup-whisper-aero-aims-to-keep-noise-levels-down

Visit Norway. (2022). Ullandhaugtårnet. Retrieved November 11, 2022, from https://www.visitnorway.no/listings/ullandhaugt%C3%A5rnet/14669/

Watkins, S., Burry, J., Mohamed, A., Marino, M., Prudden, S., Fisher, A., Kloet, N., Jakobi, T., & Clothier, R. (2020). Ten questions concerning the use of drones in urban environments. *Building and Environment*, *167*, 106458.

Weibel, R. E., & Hansman, R. J. (2015). *Safety considerations for operation of unmanned aerial vehicles in the national airspace system* (tech. rep.). MIT International Center for Air Transportation.

WHO. (2018). *Environmental noise guidelines for the European Union* (tech. rep.). World Health Organization.

Woodley, B., & Acar, L. (2004). Autonomous control of a scale model of a trailer-truck using an obstacle-avoidance path-planning hierarchy. *Proceedings of the 2004 American Control Conference*, *4*, 3399–3404 vol.4.

Wu, J., Wang, H., Li, N., Yao, P., Huang, Y., & Yang, H. (2018). Path planning for solar-powered UAV in urban environment. *Neurocomputing*, *275*, 2055–2065.

Xie, X., Tang, Z., & Cai, J. (2022). The multi-objective inspection path-planning in radioactive environment based on an improved ant colony optimization algorithm. *Progress in Nuclear Energy*, *144*, 104076. https://doi.org/https://doi.org/10.1016/j.pnucene.2021.104076

Yang, P., Tang, K., Lozano, J. A., & Cao, X. (2015). Path planning for single unmanned aerial vehicle by separately evolving waypoints. *IEEE Transactions on Robotics*, *31*(5), 1130–1146.

Yang, Q., & Yoo, S.-J. (2018). Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms. *IEEE Access*, *6*, 13671–13684.

Yao, M., & Zhao, M. (2015). Unmanned aerial vehicle dynamic path planning in an uncertain environment. *Robotica*, *33*(3), 611–621.

Yilmaz, D., & Sagsöz, A. (2011). In the context of visual pollution: Effects to Trabzon city center silhoutte. *Asian social science*, *7*(5), 98.

Yin, C., Xiao, Z., Cao, X., Xi, X., Yang, P., & Wu, D. (2018). Offline and online search: UAV multiobjective path planning under dynamic urban environment. *IEEE Internet of Things Journal*, *5*(2), 546–558.

Zaccone, R., & Martelli, M. (2018). A random sampling based algorithm for ship path planning with obstacles. *Proceedings of the International Ship Control Systems Symposium (iSCSS)*.

Zhang, J., Campbell, J. F., Sweeney II, D. C., & Hupman, A. C. (2021). Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, *90*, 102668.

Zhang, Y., Chen, J., & Shen, L. (2012). Hybrid hierarchical trajectory planning for a fixed-wing UCAV performing air-to-surface multi-target attack. *Journal of Systems Engineering and Electronics*, *23*(4), 536–552.

Zhao, Y., Zheng, Z., & Liu, Y. (2018). Survey on computational-intelligence-based UAV path planning. *Knowledge-Based Systems*, *158*, 54–64.

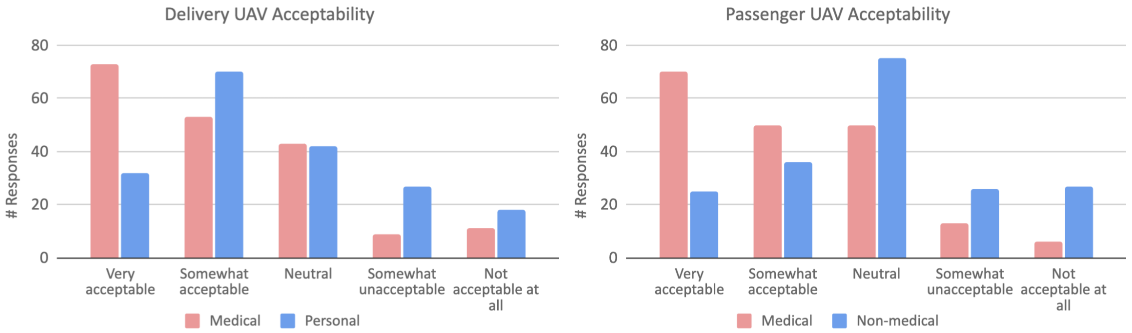
Appendices

A AiRMOUR Studies

This appendix was designed as part of our preparatory research project and presents some of the research conducted by the EU-funded AiRMOUR project. Section A.1 presents results from a survey conducted in February 2022, and Section A.2 presents findings from a demonstration held in September 2022, which we were able to participate in. Note that some terminology is changed to fit with the terminology used in the rest of this thesis, e.g. the word *drone* is replaced with the word *UAV*.

A.1 AiRMOUR Citizen Questionnaire, Norway February 2022

The AiRMOUR Citizen Questionnaire was answered by regular citizens and had almost 200 respondents from Oslo/Viken, Rogaland and Vestland. The survey concerned usage of both delivery and passenger UAVs. Figure A.1 shows that for both delivery and passenger UAVs, people are generally more positive to UAVs with medical purposes.



(a) Delivery UAV acceptance.

(b) Passenger UAV acceptance.

Figure A.1: Degree of acceptance amongst respondents for medical and personal deliveries from delivery UAVs, and medical and non-medical use of passenger UAVs.

Furthermore, the questionnaire found that about one third of the respondents strongly or somewhat agreed that they would be comfortable with a delivery or passenger UAV flying above them when walking along the street. On the other hand, 20% expressed a strong disagreement to the statement for delivery UAVs and 25% stated the same for passenger UAVs.

Figure A.2 shows the respondents' degree of concern regarding safety, visual pollution and noise pollution from UAVs in general. It is clear that people have some inherent concerns, however the answer option "Somewhat concerned" is more common than "Very concerned".

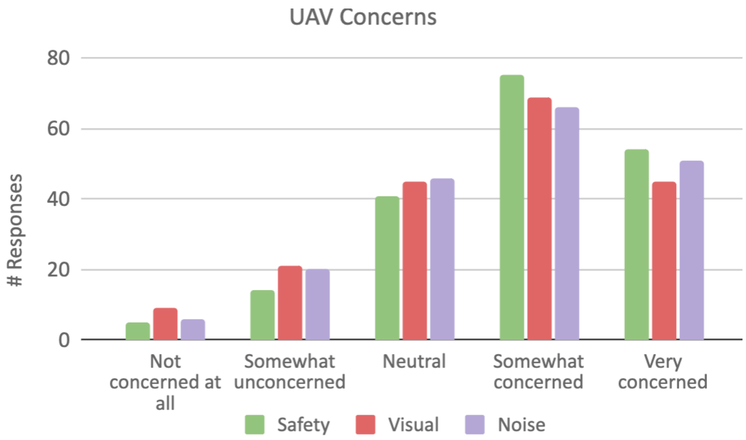
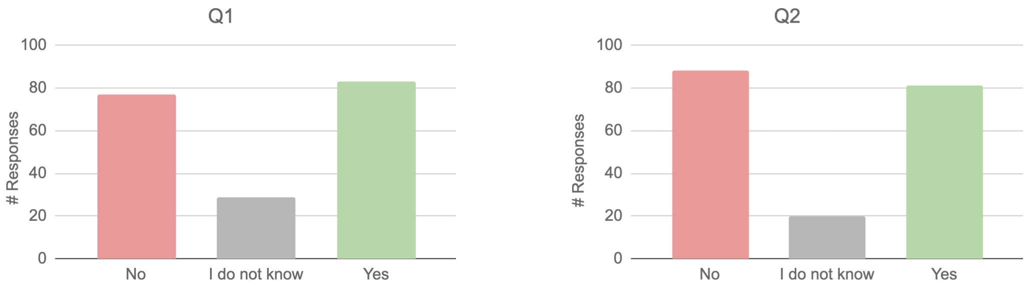


Figure A.2: Respondents' degree of concern regarding safety, visual pollution and noise pollution.

Building on the issue of noise pollution, the respondents were faced with the following two questions:

- **Q1**: If a passing delivery UAV makes the same level of noise as a passing car, would that be an issue for you?
- **Q2**: If a passing passenger UAV makes the same level of noise as a passing ambulance with its siren, would that be an issue for you?

The answers to these questions are shown in Figure A.3. The figure indicates that people have diverging attitudes towards UAV noise, with the number of "Yes" and "No" answers being fairly similar in size.



(a) Noise from delivery UAV.

(b) Noise from passenger UAV.

Figure A.3: Respondents' attitudes towards noise emitted by delivery and passenger UAVs.

The answers show that people have different attitudes towards UAV noise, and the yes and no pools are of fairly similar size.

A.2 AiRMOUR Demonstration, Stavanger September 2022

On 21st of September 2022, AiRMOUR held a UAV demonstration at Mostun Natursenter in Stavanger. Both technical stakeholders from e.g. the medical sector and Stavanger Kommune, as well as regular citizens, were invited and around 30 people attended. The Stavanger demonstration was conducted in cooperation with LuxMobility, a transport consultancy firm based in Luxembourg.

The event was divided into three parts. Before the demonstration began, the attendees were invited to take a pre-survey mapping their current attitudes towards medical use cases for UAVs. Then there was a live demonstration of a flying delivery UAV and an exhibition of a passenger UAV, followed by the attendees conducting an on-site survey about the demonstration. The last part of the event was focus group discussions where the attendees could elaborate on their thoughts and opinions, as well as if the demonstration had led to changes in their attitudes towards UAV. The findings from the demonstration event that are most relevant for this thesis are presented below.

Pre-survey

The pre-survey showed that 70% of the respondents strongly agreed that, given adequate technology, delivery UAVs should be used for urgent medical deliveries, such as for transport of organs or blood products. For non-urgent medical deliveries, the answers varied more, but almost 50% responded that they strongly or somewhat agreed to such a use case.

The respondents were also asked to state their concerns regarding the following aspects of delivery UAV usage:

- **Q1**: Safety, such as flying vehicles possibly crashing
- **Q2**: Visual pollution, such as annoying air traffic
- **Q3**: Noise pollution, such as loud and/or annoying sounds of flying aircraft

The answers are displayed in Figure A.4. Although these questions only consider delivery UAVs, they are quite similar to those displayed in Figure A.2 in Subsection A.1. Again, we see that "Somewhat concerned" is the most chosen alternative for all three aspects. However, in contrast to the AiRMOUR Citizen Questionnaire result, very few people in this survey answered that they are very concerned. A potential explanation for this is that some of the respondents were stakeholders with prior interest and knowledge on the topic.

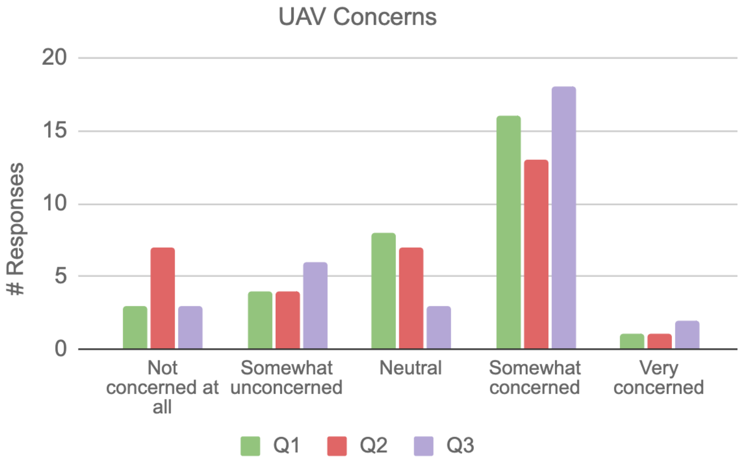


Figure A.4: Responses to survey questions Q1-Q3.

Demonstration Site Survey

During the demonstration, a delivery UAV was flown above the area where the attendees were standing, and they were asked how they rated the noise and visual impact of the UAV when it was at the highest point they could see. The responses are shown in Figure A.5. The figure indicates that the respondents in general experienced quite low noise and visual impact, with about 70% of the respondents choosing numbers 1 to 3.

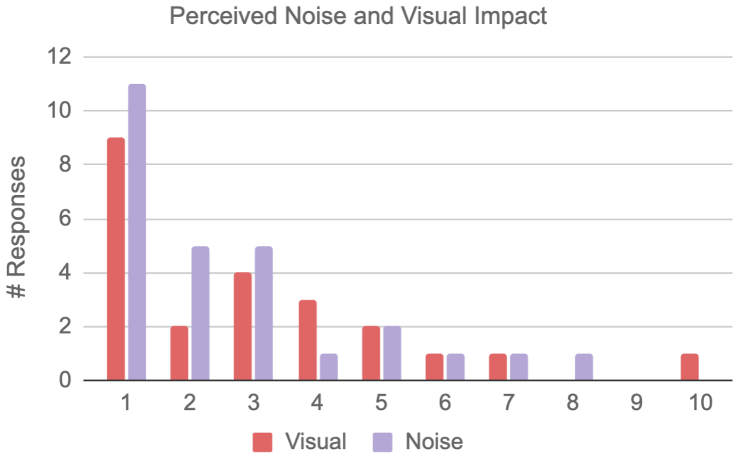


Figure A.5: Perception of UAV noise and visual impact during the demonstration, with a value of 1 indicating no impact and 10 indicating extreme impact.

Post-demonstration Focus Groups

A total of five focus group discussions were conducted, and the responses were quite similar across the groups. In general, people were more positive to using UAVs for emergency medical purposes than for non-urgent purposes. Regarding the latter, the attendees had diverging opinions. Some thought non-urgent UAV deliveries could be very convenient and that it could be especially useful for transport of e.g. pharmaceutical products to more rural areas. Others expressed great concerns in regards to privacy and surveillance. Some attendees were positive to the use of colors or other trademarks to clarify the use case of a UAV, similar to how emergency helicopters are colored yellow or ambulances have blue lights. Several attendees stated that the UAV flown in the demonstration generated less visual and noise pollution than what they expected beforehand. Lastly, it was a common attitude that strict regulations have to be in place if UAV usage is to be introduced at a large scale, and that increased information about UAV usage could help to increase acceptability for widespread usage.

B Overview of World and Objective Models

This appendix gives a structured overview of the equations for energy, risk, visual pollution and noise pollution developed in Chapter 6 and used in the MINLP in Chapter 5. Section B.1 presents the energy consumption model used in constraints (5.12) in the MINLP. Sections B.2-B.4 present the risk, visual pollution and noise pollution models, respectively. For each objective model, the single-UAV perspective is presented first and then the system perspective, with the system perspective being the model used in the objective function vector (5.1) in the MINLP. For further details on any equation, refer to Chapter 6.

B.1 Energy Consumption Model

Sets

$\mathcal{N}$ Set of vertices

$\mathcal{N}_i$ Set of neighboring vertices to vertex i, $\mathcal{N}_i \subseteq \mathcal{N}$

$\mathcal{S}_u$ Set of segments included in the voyage for UAV u

Parameters

E_u^A Energy consumed by UAV u traveling horizontally per kilogram of weight per meter above sea level per unit of time, [J/(kg·m·s)]

E_u^D Energy consumed by UAV u traveling vertically downwards per kilogram of weight per meter above sea level of the starting height per unit of time, [J/(kg·m·s)]

E_u^U Energy consumed by UAV u traveling vertically upwards per kilogram of weight per meter above sea level of the starting height per unit of time, [J/(kg·m·s)]

E_u^W Energy consumed by UAV u per kilogram of weight per unit of time when the UAV is in active mode, [J/(kg·s)]

H_i^S Altitude of vertex i above sea level, [m]

T_{uij} Time UAV u spends flying from vertex i to vertex j, [s]

W_u Weight of UAV u, [kg]

W_{us}^P Weight of the payload in UAV u on segment s, [kg]

Equations

The total energy consumption on a path for UAV u is given by

$$E_u = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} T_{uij} \cdot (W_u + W_{us}^P) \cdot (E_u^W + E_{ui}^F) \,,$$

$$\text{where} \quad E_{ui}^F = \begin{cases} H_i^S \cdot E_u^A, & \text{if } j \text{ is horizontal to } i \\ H_i^S \cdot E_u^U, & \text{if } j \text{ is above } i \\ H_i^S \cdot E_u^D, & \text{if } j \text{ is below } i \end{cases} \tag{B.1}$$

B.2 Risk Model

Sets

$\mathcal{N}$	Set of vertices
$\mathcal{U}$	Set of UAVs
$\mathcal{S}_u$	Set of segments included in the voyage for UAV u
$\mathcal{G}^R_{uijs}$	Set of ground squares that are entirely within the lethal area associated with the edge between vertex i and vertex j on segment s for UAV u
$\mathcal{N}_i$	Set of neighboring vertices to vertex i, $\mathcal{N}_i \subseteq \mathcal{N}$
$\mathcal{T}$	Set of time interval start times, in multiples of δ
$\mathcal{N}(u,\tau,\delta,s,\vec{\boldsymbol{t}})$	Set of vertices visited by UAV u on segment s in system flight plan $\vec{\boldsymbol{t}}$ in time interval $[\tau, \tau+\delta)$
$\mathcal{U}^C(u,\tau,\delta,\vec{\boldsymbol{t}})$	Set of UAVs within the collision risk area of UAV u in system flight plan $\vec{\boldsymbol{t}}$ in time interval $[\tau, \tau+\delta)$

Parameters

α	Factor for maximum reduction in mean time between falldowns due to the possibility of collisions
δ	Length of time interval regarded as simultaneous UAV impact, [s]
A^G_u	Glide angle of UAV u, [°]
C^G	Acceleration of free fall, [m/s^2]
$C^{R,Low}$	Required impact energy to cause a fatality as the sheltering factor, F_g, goes to zero, [J]
$C^{R,Mid}$	Required impact energy for a fatality probability of 50% with the sheltering factor F_g=0.5, [J]
D^{Min}	Minimum distance between two UAVs to avoid collision, [m]
D_g	Population density in ground square g, [# individuals/m^2]
$D(u,u',t,\vec{\boldsymbol{t}})$	Distance between UAV u and UAV u' in system flight plan $\vec{\boldsymbol{t}}$ at time t, [m]
F_g	Sheltering factor in ground square g
H^P	Average height of a person, [m]
H^G_i	Altitude of vertex i above ground level, [m]
K_{us}	Number of passengers in UAV u on segment s [# individuals]
M^L_u	Length of UAV u, [m]
T^F_u	Mean time between failures for UAV u, [s]
T_{uij}	Time UAV u spends flying from vertex i to vertex j, [s]
V_u	Speed of UAV u, [m/s]
W_u	Weight of UAV u, [kg]

W_u^{Max}	Weight baseline in relation to risk for UAV u, [kg]
W_{us}^P	Weight of the payload in UAV u on segment s, [kg]

Single-UAV Perspective Equations

The total risk associated with UAV u flying a path is given by

$$R_u = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} \frac{1}{T_u^F} \cdot (N_{us}^A \cdot B^A + N_{uijs}^G \cdot B_{uijs}^G) \cdot T_{uij} \cdot x_{uijs} \,,$$

$$\text{where} \quad N_{us}^A = K_{us} \,, \quad B^A = 1 \,, \quad N_{uijs}^G = (W_u + W_{us}^P) \cdot \left(M_u^L + \frac{H^P}{sin(A_u^G)} \right) \cdot \frac{1}{|\mathcal{G}_{uijs}^R|} \sum_{g \in \mathcal{G}_{uijs}^R} D_g \,,$$

$$B_{uijs}^G = \frac{1}{|\mathcal{G}_{uijs}^R|} \sum_{g \in \mathcal{G}_{uijs}^R} \frac{1}{1 + \sqrt{\frac{C^{R,Mid}}{C^{R,Low}}} \cdot \left(\frac{C^{R,Low}}{\frac{1}{2} \cdot (W_u + W_{us}^P) \cdot (C^G \cdot (H_i^G + H_j^G) + (V_u)^2)} \right)^{\frac{1}{4F_g}}} \tag{B.2}$$

System Perspective Equations

The total risk associated with system flight plan $\vec{\boldsymbol{t}}$ is given by

$$R(\vec{\boldsymbol{t}}) = \sum_{\tau \in \mathcal{T}} \sum_{u \in \mathcal{U}} \sum_{\substack{i \in \\ \mathcal{N}(u,\tau,\delta,s,\vec{\boldsymbol{t}})}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} \frac{1}{T_u^F - T^C(u, \tau, \delta, \vec{\boldsymbol{t}})} \cdot (N_{us}^A \cdot B^A + N_{uijs}^G \cdot B_{uijs}^G) \cdot T_{uij} \cdot x_{uijs} \,,$$

where

$$T^C(u, \tau, \delta, \vec{\boldsymbol{t}}) = \alpha \cdot T_u^F \cdot min \left(\sum_{\substack{u' \in \\ \mathcal{U}^C(u,\tau,\delta,\vec{\boldsymbol{t}})}} \frac{\frac{W_{u'}}{W_u^{Max}}}{\min(D(u, u', \tau, \vec{\boldsymbol{t}}), D(u, u', \tau + \delta, \vec{\boldsymbol{t}})) - D^{Min} + 1}, 1 \right) \tag{B.3}$$

B.3 Visual Pollution Model

Sets

$\mathcal{G}$	Set of ground squares
$\mathcal{N}$	Set of vertices
$\mathcal{U}$	Set of UAVs
$\mathcal{G}^V_{ui}$	Set of affected ground squares when UAV u is positioned in vertex i
$\mathcal{N}_i$	Set of neighboring vertices to vertex i, $\mathcal{N}_i \subseteq \mathcal{N}$
$\mathcal{S}_u$	Set of segments included in the voyage for UAV u

Parameters

β	Discount factor for the system perspective visual pollution model
δ	Length of time interval regarded as simultaneous UAV impact, [s]
$C^{V,1}$	Constant used in the visual pollution model
$C^{V,2}$	Constant used in the visual pollution model
D^E	Distance between ruler and eye when measuring perceived UAV size, [m]
D^U_{ig}	Distance between a UAV located in vertex i and the center of ground square g, [m]
F_g	Sheltering factor in ground square g
I^F_{uis}	1 if vertex i is the end vertex of segment s for UAV u, 0 otherwise
M^L_u	Length of UAV u, [m^2]
Q_g	Population in ground square g, [# individuals]
R^V	Threshold for visual pollution, [m]

Single-UAV Perspective Equations

The total visual pollution generated by UAV u flying a path is given by

$$\begin{aligned} P^V_u &= \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}_i} \sum_{s\in\mathcal{S}_u} P^V_{ui} \cdot x_{uijs} + \sum_{i\in\mathcal{N}} \sum_{s\in\mathcal{S}_u} P^V_{ui} \cdot I^F_{uis} \;, \\ \text{where } P^V_{ui} &= \sum_{g\in\mathcal{G}^V_{ui}} (1-F_g)\cdot(C^{V,1}\cdot \frac{1}{(D^U_{ig})^{C^{V,2}}})\cdot Q_g \end{aligned} \tag{B.4}$$

The ground squared included in the set of affected ground squares for UAV u have their centers within a sphere with radius D^V_u

$$D^V_u = \frac{M^L_u}{R^V} \cdot D^E \tag{B.5}$$

System Perspective Equations

The total visual pollution generated by system flight plan $\vec{t}$ is given by

$$P^V(\vec{t}) = \sum_{g\in\mathcal{G}} P^V(\vec{t}, g) \tag{B.6}$$

with $P^V(\vec{t}, g)$ given by Algorithm 7. In line 8 of the algorithm, $D(t_{uis}, g)$ is a function returning the same value as D^U_{ig}.

Algorithm 7 Algorithm to find the total visual pollution generated by a system flight plan in a given ground square g

```
    Input: System flight plan $\vec{t}$, ground square $g$
    Output: Visual pollution in ground square $g$, $P^V_g$
 1: function $P^V(\vec{t}, g)$
 2:     $P^V_g \leftarrow 0$
 3:     $t^{Max} \leftarrow \max\{t_{uis} \in \vec{t}\} - \min\{t_{uis} \in \vec{t}\}$
 4:     $\mathcal{T} \leftarrow \{0, \delta, 2\delta, ..., \lfloor \frac{t^{Max}}{\delta} \rfloor \delta\}$
 5:     for $\tau \in \mathcal{T}$ do
 6:         $P^V_{g\tau} \leftarrow []$
 7:         for $\{t_{uis} \in \vec{t} \mid \exists_{u\in\mathcal{U},\, i\in\mathcal{N},\, s\in\mathcal{S}_u}\, t_{uis} \in [\tau, \tau+\delta) \wedge g \in \mathcal{G}^V_{ui}\ \}$ do
 8:             $P^V_{g\tau}.append\left((1-F_g)\cdot(C^{V,1}\cdot\frac{1}{D(t_{uis},g)^{C^{V,2}}})\cdot Q_g\right)$
 9:         end for
10:         $P^V_{g\tau} \leftarrow DescendingSort(P^V_{g\tau})$
11:         for $n \in \{0, 1, ..., |P^V_{g\tau}| - 1\}$ do
12:             $P^V_g \leftarrow P^V_g + P^V_{g\tau}[n]\cdot\beta^n$
13:         end for
14:     end for
15:     return $P^V_g$
16: end function
```

B.4 Noise Pollution Model

Sets

$\mathcal{G}$ Set of ground squares

$\mathcal{N}$ Set of vertices

$\mathcal{U}$ Set of UAVs

$\mathcal{G}^N_{ui}$ Set of affected ground squares when UAV u is positioned in vertex i with threshold R^N

$\mathcal{G}^{NS}_{ui}$ Set of affected ground squares when UAV u is positioned in vertex i with threshold R$^N - R^{N,Red}$

$\mathcal{N}_i$ Set of neighboring vertices to vertex i, $\mathcal{N}_i \subseteq \mathcal{N}$

$\mathcal{S}_u$ Set of segments included in the voyage for UAV u

Parameters

δ Length of time interval regarded as simultaneous UAV impact, [s]

A^U_{ig} Angle between a UAV located in vertex i, the center of ground square g and the horizontal, [°]

A^M_u Angle between UAV u, measurement device and the horizontal when SPL was measured, [°]

D^U_{ig} Distance between a UAV located in vertex i and the center of ground square g, [m]

D^M_u Distance between UAV u and measurement device when SPL was measured, [m]

M^S_u SPL of UAV u measured by measurement device, [dB]

O Decrease in SPL per degree decrease in angle between the vertex the UAV is located in, measurement device and the horizontal, [dB/°]

p_0 Reference level for sound pressure, [Pa]

Q_g Population in ground square g, [# individuals]

R^N Threshold for noise pollution, [dB]

$R^{N,Red}$ Reduction in threshold for noise pollution to find affected ground squares in the system perspective, [dB]

Single-UAV Perspective Equations

The total noise pollution generated by UAV u flying a path is given by

$$P^N_u = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}_i} \sum_{s \in \mathcal{S}_u} P^N_{ui} \cdot x_{uijs} + \sum_{i \in \mathcal{N}} \sum_{s \in \mathcal{S}_u} P^N_{ui} \cdot I^F_{uis} \,,$$

$$\text{where } P^N_{ui} = \sum_{g \in \mathcal{G}^N_{ui}} 2^{\frac{M^S_u + O\cdot(90 - A^M_u) + 20\cdot\log_{10}(D^M_u) - O\cdot(90 - A^U_{ig}) - 20\cdot\log_{10}(D^U_{ig})}{10}} \cdot Q_g \tag{B.7}$$

The cut-off distance, D_u^N, used to find potentially affected ground squares is given by

$$D_u^N = 10^{\frac{M_u^S + O\cdot(90 - A_u^M) + 20\cdot\log_{10}(D_u^M) - R^N}{20}} \tag{B.8}$$

The ground squared included in the set of affected ground squares for UAV u have their centers within a sphere with radius D_u^N and SPL value exceeding the threshold, R^N

$$S^M + O\cdot(90 - A^M) + 20\cdot\log_{10}(D^M) - O\cdot(90 - A_{ig}^U) - 20\cdot\log_{10}(D_{ig}^U) \geq R^N \tag{B.9}$$

System Perspective Equations

The total noise pollution generated by system flight plan $\vec{\boldsymbol{t}}$ is given by

$$P^N(\vec{\boldsymbol{t}}) = \sum_{g\in\mathcal{G}} P^N(\vec{\boldsymbol{t}}, g) \tag{B.10}$$

with $P^N(\vec{\boldsymbol{t}}, g)$ given by Algorithm 8. In line 8 of the algorithm, $A(t_{uis}, g)$ is a function returning the same value as A_{ig}^U, and $D(t_{uis}, g)$ is a function returning the same value as D_{ig}^U.

Algorithm 8 Algorithm to find the total noise pollution generated by a system flight plan in a given ground square g

```
    Input: System flight plan $\vec{\boldsymbol{t}}$, Ground square $g$
    Output: Noise pollution in ground square $g$, $P_g^N$
 1: function $P^N(\vec{\boldsymbol{t}},\ g)$
 2:     $P_g^N \leftarrow 0$
 3:     $t^{Max} \leftarrow \max\{t_{uis} \in \vec{\boldsymbol{t}}\} - \min\{t_{uis} \in \vec{\boldsymbol{t}}\}$
 4:     $\mathcal{T} \leftarrow \{0, \delta, 2\delta, ..., \lfloor\frac{t^{Max}}{\delta}\rfloor\delta\}$
 5:     for $\tau \in \mathcal{T}$ do
 6:         $P_{gt}^N \leftarrow 0$
 7:         for $\{t_{uis} \in \vec{\boldsymbol{t}} \mid \exists_{u\in\mathcal{U},\, i\in\mathcal{N},\, s\in\mathcal{S}_u}\ t_{uis} \in [\tau, \tau+\delta) \wedge g \in \mathcal{G}_{ui}^{NS}\ \}$ do
 8:             $P_{g\tau}^N \leftarrow P_{g\tau}^N + \left(p_0 \cdot 10^{\frac{M_u^S + O\cdot(90 - A_u^M) + 20\cdot\log_{10}(D_u^M) - O\cdot(90 - A(t_{uis},g)) - 20\cdot\log_{10}(D(t_{uis},g))}{20}}\right)^2$
 9:         end for
10:         $SPL_g \leftarrow 20\cdot\log_{10}\left(\frac{\sqrt{P_{g\tau}^N}}{p_0}\right)$
11:         if $SPL_g > R^N$ then
12:             $P_g^N \leftarrow P_g^N + 2^{\frac{SPL_g}{10}} \cdot Q_g$
13:         end if
14:     end for
15:     return $P_g^N$
16: end function
```

C Multi-Objective Dijkstra's Algorithm

This appendix gives a more detailed explanation of the various methods used in the MDA, presented in Section 7.2. Sections C.1 present the $getNextCandidateLabel$ method, and Section C.2 presents the $propagate$ method. In the algorithms for both methods, we use the term *set domination*, denoted $\mathcal{Z} \preceq z'$, to mean that there exists at least one label $z \in \mathcal{Z}$ that Pareto dominates or has an equal cost vector to a label z'.

C.1 Selection of the Next Candidate Label

The $getNextCandidateLabel$ method is a part of the search strategy used by the MDA and is shown in Algorithm 9. The method creates a new candidate label for vertex v by finding the lexicographically smallest efficient label, z_w, where w is a neighbor of v, and extending this label.

Algorithm 9 begins with an empty candidate label with all objective costs set to infinity in line 2. Then lines 3-25 check the ingoing arc (w, v) for every neighbor w of v to decide whether z_w should be extended to create the next candidate label for v. In lines 4-8, the costs associated with the arc (w, v) are retrieved and the arc is added to the last processed label set $\mathcal{D}$ if an index for (w, v) does not already exist. Lines 9-24 check the efficient labels in the set $\mathcal{Z}_w$ from the index of the last processed label to the end of $\mathcal{Z}_w$. This is done by extending the label z_w^i to a potential candidate label z_v^p and checking for dominance and lexicographical ordering. The reason why only the last $|\mathcal{Z}_w|$ - i labels in $\mathcal{Z}_w$ need to be checked to find a candidate label, is due to the lexicographic order of $\mathcal{Z}_w$. This ordering is a consequence of the fact that a label z_u can only be added to its efficient set $\mathcal{Z}_u$ from the priority queue, $\mathcal{K}$. Since the priority queue always stores the lexicographically smallest candidate label z_u for a given vertex u, the efficient label set $\mathcal{Z}_u$ inherits this property. Maristany de las Casas et al. (2021) argue that because of the lexicographical ordering of the efficient label sets, if the labels prior to position i in an efficient label set $\mathcal{Z}_w$ was not chosen as the base for a candidate label in vertex v in the $(j-1)$th search, they will never be chosen in the jth or later searches. Therefore, the for-loop can start from the index of the last processed label in $\mathcal{Z}_w$, as shown in line 9, instead of checking all the labels $\mathcal{Z}_w$ for every run of the $getNextCandidateLabel$ algorithm. The lexicographical ordering of the efficiency set also enables the algorithm to break the for-loop in line 19 when a candidate label z_v^c from efficient label set $\mathcal{Z}_w$ at position i is found, because all the labels in positions $> i$ are lexicographically larger than z_v^c. Thus, in any search for a candidate label, the number of labels that must be tested for dominance is reduced. The lower bound pruning described in Section 7.2 is applied in lines 12-13. Line 15 in Algorithm 9 allows for additional pruning by checking if the extended label z_v^p is dominated by any labels in the efficient label set of the end vertex $\mathcal{Z}_f$. This line only applies when $\mathcal{Z}_f$ is not empty, meaning that the algorithm has

Algorithm 9 Algorithm to find the next candidate label to add to the priority queue for a vertex v

```
Input: Vertex $v$, last processed label set $\mathcal{D}$, total set of efficient labels $\mathcal{Z}$, set of lower
       bound cost vectors $\underline{\mathcal{C}}$
Output: Candidate label $z_v^c$
function GetNextCandidateLabel($v$, $\mathcal{D}$, $\mathcal{Z}$, $\underline{\mathcal{C}}$)
    $z_v^c \leftarrow (None, (\infty, ..., \infty), 0, None)$
    for $w \in getNeighbors(v)$ do
        $e \leftarrow getEnergyCost(w, v)$
        $c \leftarrow getEdgeCost(w, v)$
        if $(w, v) \notin \mathcal{D}$ then
            $\mathcal{D}[(w, v)] \leftarrow 0$
        end if
        for $i \in \{\mathcal{D}[(w, v)], \mathcal{D}[(w, v)] + 1, ..., |\mathcal{Z}_w|\}$ do
            $\mathcal{D}[(w, v)] \leftarrow i$
            $z_w^i \leftarrow \mathcal{Z}_w[i]$
            $z_f^{\underline{c}} \leftarrow (f, c(z_w^i) + c + \underline{c}_v, None, None)$
            if not $\mathcal{Z}_f \preceq z_f^{\underline{c}}$ then
                $z_v^p \leftarrow (v, c(z_w^i) + c, e(z_w^i) + e, p(z_w^i).append(v))$
                if not $\mathcal{Z}_f \preceq z_v^p$ then
                    if not $\mathcal{Z}_v \preceq z_v^p$ then
                        if $z_v^p$ lexicographically smaller than $z_c^p$ then
                            $z_v^c \leftarrow z_v^p$
                            break
                        end if
                    end if
                end if
            end if
        end for
    end for
    if $c(z_v^c) = (\infty, ..., \infty)$ then
        return $None$
    end if
    return $z_v^c$
end function
```

reached the end vertex f. This implies that the algorithm possibly can dominate away a lot of labels as soon as it has reached the end vertex. Unfortunately, this advantage will not apply before the algorithm has reached the end vertex for the first time. If no candidate label is found in the for-loop in lines 3-25, *None* is returned in line 27. Otherwise the next candidate label is returned in line 29.

C.2 Propagation

The *Propagate* method is another part of the search strategy used by the MDA and is shown in Algorithm 10. The method extends the label z_v for a vertex v along an outgoing edge (v, w), and potentially updates the priority queue for the vertex w.

Algorithm 10 Algorithm to propagate a label z_v along the outgoing edge (v, w)

```
    Input: Graph $G = (\mathcal{V}, \mathcal{E})$, label $z_v$, neighbor vertex $w$, priority queue $\mathcal{K}$, total set of efficient
        labels $\mathcal{Z}$, set of lower bound cost vectors $\underline{\mathcal{C}}$
    Output: Potentially updated priority queue $\mathcal{K}$
 1: function PROPAGATE($G$, $z_v$, $w$, $\mathcal{K}$, $\underline{\mathcal{C}}$)
 2:     $e \leftarrow getEnergyCost(v, w)$
 3:     if not $isEnergyFeasible(e(z_v) + e)$ then
 4:         return $\mathcal{K}$
 5:     end if
 6:     $z_f^{\underline{c}} \leftarrow (f,\ c(z_v) + c + \underline{c}_w,\ None,\ None)$
 7:     if not $\mathcal{Z}_f \preceq z_f^{\underline{c}}$ then
 8:         $c \leftarrow getEdgeCost(v, w)$
 9:         $z_w \leftarrow (w,\ c(z_v) + c,\ e(z_v) + e,\ p(z_v).append(w))$
10:         if not $\mathcal{Z}_f \preceq z_w$ then
11:             if not $\mathcal{Z}_w \preceq z_w$ then
12:                 if $w \notin \mathcal{K}$ then
13:                     $\mathcal{K} \leftarrow \mathcal{K}.insert(z_w)$
14:                 else if $z_w$ lexicographically smaller than $k_w$ then
15:                     $\mathcal{K} \leftarrow \mathcal{K}.remove(w)$
16:                     $\mathcal{K} \leftarrow \mathcal{K}.insert(z_w)$
17:                 end if
18:             end if
19:         end if
20:     end if
21:     return $\mathcal{K}$
22: end function
```

Algorithm 10 begins by computing the energy consumption for (v, w) in line 2. In lines 3-5, the algorithm checks if the energy consumption of the extended label exceeds the maximum energy consumption, thus making the path infeasible. In that case there is no need to propagate further, and the priority queue is returned with no modifications. If the extended label is energy feasible, lower bound pruning is performed in lines 6-7. If the lower bound cost label is not dominated or equaled, the actual label is extended along the edge (v, w), as shown in lines 8-9. Line 10 checks if the extended label z_w is dominated by the efficient label set of the end vertex $\mathcal{Z}_f$, and line 11 checks if z_w is dominated by the efficient label set of the vertex w, $\mathcal{Z}_w$. If any of these apply, z_w should not be added to the priority queue $\mathcal{K}$. However, if z_w is not dominated, the label is potentially a part of a Pareto optimal path. Since the priority queue $\mathcal{K}$ at any point in the MDA only should store the lexicographically smallest candidate label for a given vertex, the next step is to check whether or not vertex w is in the priority queue $\mathcal{K}$, and modify $\mathcal{K}$ accordingly. This is done in lines 12-17. Lastly, the algorithm returns the modified priority queue $\mathcal{K}$ in line 21.

D Overview of Input Parameters

This appendix presents the input parameters used in the computational study, and is partially based on work done in our preparatory research project. Section D.1 describes the parameters of the urban environment, Section D.2 explains the UAV parameters, Section D.3 details the objective model parameters and Section D.4 presents the parameters used in the HEA solution method. Note that this appendix uses the same naming convention as in the computational study; HEA^{C} denotes the complete algorithm, HEA^{F} refers to running HEA with $isSystem = False$ and HEA^{T} refers to all remaining parts of HEA^{C} when omitting HEA^{F}. Also note that the values for the parameters that are modified in the computational study are explained in Section 8.1, and they are thus not explained again in this appendix.

D.1 Urban Environment Parameters

Table D.1 shows the parameters used to create the urban environment. While the first four parameters are independent of the environment, the minimum and maximum longitudes and latitudes depend on the chosen environment. The computational study uses Stavanger, a city on the south-west coast of Norway, as a case study. We chose an actual city, as opposed to a fictional one, to get a more realistic computational study and to simplify the information gathering, as we do not possess enough prior knowledge to design a realistic city from scratch. Stavanger is the third largest city in Norway by population (Statistisk Sentralbyrå, 2021), and a test arena for emergency medical UAVs by the AiRMOUR project (Stavanger kommune, n.d.). The outermost coordinates of the environment used correspond to an area of 4.97 km west-east and 2.60 km south-north. The area is the most urban part of Stavanger, and includes the city center, a hospital and nature areas such as parks and lakes. Note that any urban environment, real or fictional, could have been used as the basis for the computational study in this thesis, and the solution method does not depend on the city of Stavanger in any way.

The city of Stavanger is discretized into a graph for the UAVs to fly in, with the length of a gridline between gridline intersections, L, set relatively small to obtain more realistic and smooth UAV paths. Data about no-flight zones is obtained using the map from Safe To Fly (n.d.), which collects data from different external sources to provide a complete map of no-flight zones in Norway. On the ground, there is a separate set of ground squares. Each ground square is assigned an altitude above sea level, a population and a sheltering factor. Data about altitudes above sea level is found by combining coordinate data from OpenStreetMap (n.d.) with altitude data from Topographic Map (n.d.). To assign ground squares with a population Q_g, we use data from Statistisk Sentralbyrå (2022), containing the population for squares with a side length 250 meters. The population is distributed evenly for all the ground squares within each 250x250 meter square. Topological data for the sheltering factor is obtained from OpenStreet-

Table D.1: Overview of parameters values for the urban environment.

Parameter [unit]	Description	Value
λ	Ground square side length multiple of L	10
$H^{G,Max}$ [m]	Maximum allowed flight altitude above ground	120
$H^{G,Min}$ [m]	Minimum allowed flight altitude above ground	60
L [m]	Length of gridline between gridline intersections	10
x^{Max} [°]	Maximum longitude in environment	5.767408712025858
x^{Min} [°]	Minimum longitude in environment	5.680502674393980
y^{Max} [°]	Maximum latitude in environment	58.974277855489525
y^{Min} [°]	Minimum latitude in environment	58.950793750000000

Map (n.d.). This source contains data as a set of coordinate points with an associated area category, e.g. buildings such as schools, malls and residential homes, and recreational areas such as parks, beaches and market squares. The sheltering factor F_g is then assigned using the classification for various urban areas displayed in Table D.2, which we developed in our preparatory research project based on work by Primatesta et al. (2020).

Table D.2: Overview of sheltering factor classification in different areas.

Area characteristics	Sheltering factor, F_g
Open area, no shelter	0.01
Small constructions, e.g. public toilets or telephone booths	0.15
Small buildings, e.g. garages or gazebos	0.30
Dense forests or large constructions, e.g. bridges	0.50
Residential buildings	0.60
Industrial or public buildings	0.90

The minimum allowed flight altitude above ground, $H^{G,Min}$, is set to 60 meters, based on conversations with an EHang representative during an AiRMOUR demonstration in Stavanger, discussed in Appendix A.2. The value of the maximum allowed flight altitudes above ground, $H^{G,Max}$, is based on the UAV regulations applicable in Norway as of May 2023 (European Commission, 2019a). The regulations state that a UAV cannot fly above 120 meters from the closest point on the Earth's surface, unless it is flying over an artificial obstacle. In such cases, the UAV is allowed to fly 15 meters above the obstacle within a horizontal distance of 50 meters from the obstacle. In the specific case study of Stavanger, the last part of the regulation does not apply as the tallest building in Stavanger is 78 meters tall (Visit Norway, 2022). Thus, the maximum flight altitude is set to 120 meters everywhere in the graph.

D.2 Unmanned Aerial Vehicle Parameters

The UAV parameters are presented in Table D.3. The two UAVs used in the computational study are a passenger UAV based on the EHang 216 Autonomous Aerial Vehicle (EHang, 2016) and a delivery UAV based on the Amazon Prime Air UAV (Amazon, 2022). Many of the parameters of the Amazon Prime Air UAV are based on the analysis done by Jung and Kim (2017). Since only two UAV types are used, the UAV index u is omitted from the parameters in Table D.3.

The weight, W, the length, M^L, and the speed, V, of the UAVs are set using information about the above-mentioned UAVs. W^{Max} is set to 100 kg and 500 kg for the delivery and passenger UAV, respectively, which we consider fair values based on the weights of the two UAV types. As described in Section 2.5, the mean time between failures, T^F, varies depending on e.g. the weight, battery capacity and size of the UAV. In this thesis the value of T^F is based on the failure analysis performed by Lin and Shao (2020) on UAVs with vertical takeoff and landing, and assumed equal for the two UAV types. For simplicity, the glide angle, A^G, is set to 90° for both UAV types, i.e. the UAVs fall vertically straight down in case of malfunction or collision. The service time T_i^O is set to 30 seconds for both UAV types for all vertices i. We consider this to be a fair estimate for delivery UAVs taking into account their speed and the fact that all locations are defined to be at the minimum allowed flight altitude above ground, resulting in equal vertical traversal distances. For simplicity, we use the same value for the passenger UAVs, although in reality, potential boarding and disembarking is likely to take more time.

As described in Section 2.1, energy consumption depends on a number of internal and external factors. Thus, the energy specific parameters vary greatly depending on UAV characteristics. The energy capacity, E^{Max}, is set based on values of the Amazon Prime Air UAV and the EHang 216 UAV. It is challenging to obtain information about the remaining energy specific parameters, and we make the simplification that the weight of the UAV is the only UAV-specific factor affecting these parameters. Consequently, the remaining energy parameters are equal for the two types of UAVs, since they are given in a per-weight unit. The value of the energy consumed per kilogram per unit of time when the UAV is in active mode, E^W, is based on an average between the energy consumed for the Amazon UAV at maximum speed and when hovering (Jung & Kim, 2017). As stated in Section 2.1, energy consumption increases with flight altitude. For simplicity, we assume a linear relationship between the energy consumed and each additional meter above sea level. The value of the energy consumed traveling horizontally per kilogram per meter above sea level per unit of time, E^A, is obtained from Majeed et al. (2020). To obtain the energy consumed traveling an edge vertically downwards, E^D, and

Table D.3: Overview of parameter values for UAV characteristics. The UAV index u is omitted in the leftmost column in the table since only one type of delivery UAV and one type of passenger UAV are used in the computational study.

Parameter [unit]	**Description**	**Delivery**	**Passenger**
A^G [°]	Glide angle of the UAV	90	90
A^M [°]	Angle between the UAV, the measurement device and the horizontal when the SPL was measured	30	90
D^M [m]	Distance between the UAV and the measurement device when the SPL was measured	1	10
E^A [J/(kg·m·s)]	Energy consumed traveling horizontally per kilogram of weight per meter above sea level per unit of time	0.3237	0.3237
E^D [J/(kg·m·s)]	Energy consumed traveling vertically downwards per kilogram of weight per meter above sea level of the starting height per unit of time	0.3237	0.3237
E^{Max} [J]	Energy capacity of the UAV	$1.322 \cdot 10^6$	$91.8 \cdot 10^6$
E^U [J/(kg·m·s)]	Energy consumed traveling vertically upwards per kilogram of weight per meter above sea level of the starting height per unit of time	1.619	1.619
E^W [J/(kg·s)]	Energy consumed per kilogram of weight per unit of time when the UAV is in active mode	553.3	553.3
M^L [m]	UAV length	1.15	5.64
M^S [dB]	SPL of the UAV measured by measurement device	90	90
T^F [h]	Mean time between failures for the UAV	100	100
T^O* [s]	UAV service time	30	30
V [m/s]	UAV speed	16.67	27.78
W [kg]	UAV weight	3.8	360
W^{Max} [kg]	Weight baseline in relation to risk for the UAV	100	500

*Vertex index i omitted since the parameter value is the same for all vertices.

upwards, E^U, per kilogram per meter above sea level of the starting altitude of the edge per unit of time, we use the fact that it was found to be five times more energy consuming to fly vertically upwards than downwards (Abeywickrama et al., 2018). E^D is assumed to equal the energy consumption of flying horizontally in the starting altitude of the edge, i.e. $E^D = E^A$. Hence, $E^U = 5 \cdot E^D = 5 \cdot E^A$.

Similar to the energy parameters, there was hardly any specific data available concerning the noise pollution of different types of UAVs. For the passenger UAV, the sound pressure level measured by a measurement device, M^S, is set equal to the values from a measurement of the EHang 216 UAV (Transport Up, 2019). However, angle and distance information about the measurement was not stated. Therefore, the angle between the UAV, the measurement device and the horizontal, A^M, and the distance between the UAV and the measurement device, D^M, are assigned values we found reasonable based on common sense and some test calculations with different values of the parameters. For the delivery UAV, the values of M^S, A^M and D^M are based on noise measurements of two different multicopters, DJI Inspire 2 and DJI S-900 (Schäffer, Pieren, Heutschi et al., 2021). When assigning these values, we have ensured a plausible relationship between the noise of the passenger and the delivery UAVs.

D.3 Parameters of the Objectives

Table D.4 gives an overview of the parameters of the objectives. To estimate the lethal area of the UAVs in the risk model, we need, in addition to data about UAV characteristics, the payload and the glide angle, the average height of a person, H^P. H^P is set to 1.735 meters, which corresponds to the average height of a Norwegian (Statistisk Sentralbyrå, 2012). The values for $C^{R,Low}$ and $C^{R,Mid}$ are set using the work by Dalamagkidis et al. (2008).

In the visual pollution model, $C^{V,1}$ and $C^{V,2}$ take the same values as in the study by Thomas (2022), which the visual pollution model uses as a starting point. To calculate the cut-off distance for visual pollution D_u^V, we have to set a parameter value of D^E, which denotes the distance from an observer's eye to a measurement device used to measure the size of a flying UAV. We set the value of D^E to 0.5 meters, corresponding to the observer measuring a UAV with a ruler at an arm's length distance. D_u^V is found by taking the minimum of the calculated distance from equation (6.7) and 1000 meters. This is due to the fact that for distances exceeding 1000 meters, we assume that there will be buildings or other objects that block the view of the UAV for a potential observer.

For the noise pollution model, we need to set the value of O, which denotes the decrease in sound pressure level per degree decrease in the angle between the UAV, the measurement device and the horizontal. As described in Section 2.7, the sound pressure level decreases almost linearly with change in angle. By considering empirical research on UAVs at different frequencies and choosing a midpoint (Heutschi et al., 2020; Treichel & Körper, 2019), the sound pressure level can be said to change with approximately 8 dB depending on the angle

Table D.4: Overview of parameter values needed for calculation of the objectives.

Parameter [unit]	Description	Value
α	Factor for maximum reduction in mean time between falldowns due to the possibility of collisions	0.5
β	Discount factor for the system perspective visual pollution model	0.7
δ [s]	Length of time interval regarded as simultaneous UAV impact	5
C^G [m/s^2]	Acceleration of free fall	9.81
$C^{R,Low}$ [J]	Required energy at impact to cause a fatality as the sheltering factor, F_g, goes to zero	10^6
$C^{R,Mid}$[J]	Required energy at impact for a fatality probability of 50% with the sheltering factor F_g=0.5	100
$C^{V,1}$	Constant used in the visual pollution model	47.757
$C^{V,2}$	Constant used in the visual pollution model	0.678
D^E [m]	Distance between ruler and eye when measuring perceived UAV size	0.5
D^{Max} [m]	Collision risk threshold	45
D^{Min} [m]	Minimum distance between two UAVs to avoid collision	15
H^P [m]	Average height of a person	1.735
O [dB/°]	Decrease in SPL per degree decrease in angle between the vertex the UAV is located in, the measurement device and the horizontal	0.09
p_0 [Pa]	Reference level for sound pressure	$2 \cdot 10^{-5}$
R^N [dB]	Threshold for noise pollution	55
$R^{N,Red}$ [dB]	Reduction in threshold for noise pollution to find affected ground squares in system perspective	10
R^V [m]	Threshold for visual pollution	0.005

A^M, when A^M is between 0° and 90°. For modeling purposes, we can hence say that the sound pressure level decreases by $\frac{8\,\text{dB}}{90°} = 0.09$ dB per degree decrease in the angle A^M. $R^{N,Red}$, which is used to find the affected ground squares in the system perspective, is set to 10 dB. $R^{N,Red}$ should be set high enough to ensure that the set of affected ground squares includes all ground squares that experience noise pollution above R^N when considering the combined SPL of the UAVs in the system. At the same time, a lower value of $R^{N,Red}$ speeds up the solution method, because it results in less calculation. As described in Section 6.5, a 10 dB increase is experienced as twice as loud. With the equations presented in Section 6.5, one would need 10 UAVs with the same SPL within the same time interval to make the simultaneous noise pollution from the UAVs 10 dB higher, and we hence believe setting $R^{N,Red}$ to 10 dB is a fair value.

D.4 Solution Method Parameters

The HEA solution method contains several parameters that have to be set, both in relation to the NSGA-III and the ACO. Table D.5 shows the values selected for these parameters for HEA^{T} and HEA^{F}. The parameters of the ACO have a high degree of interdependence, exemplified by the pheromone decay rate φ and the number of ants A. A high pheromone decay rate is needed to ensure diversity in the paths found by the ants if the number of ants are low. With a large number of ants, diversity is already covered and thus a lower value for the decay rate is sufficient. Thus, the values of the ACO parameters cannot be decided in isolation, and we have found suitable values by systematically testing various combinations of the parameters to strike a balance between solution quality and runtime efficiency. We have looked at the parameter values proposed by Colorni et al. (1991), Dorigo and Gambardella (1997), Ntakolia and Lyridis (2022) and Xie et al. (2022) for guidance.

In contrast to the ACO, the NSGA-III parameters are fairly independent. We have set quite low values for the percentage improvement thresholds, c^{Min}, to ensure that the algorithm continues as long as it can make significant improvements, but without the need for complete convergence, to keep the runtime manageable. The percentage has been set higher for HEA^{F} than HEA^{T}, since HEA^{F} is only used as a preprocessing step. The higher percentage criteria for HEA^{F} also makes it natural to set a higher minimum number of iterations, I^{Min}, for this part of HEA^{C}. The preprocessing aspect is also the reason why the population size, N, for HEA^{F} has been set smaller than for HEA^{T}, and why the number of flight plans to return, M, is set equal to N for HEA^{F}. The number of objective axis divisions d for the reference point generation have been set based on values proposed by Deb and Jain (2014). Since we set $M = 5$ for HEA^{T}, the five system flight plans returned from HEA^{C} correspond to the ones with minimum values for each objective, in addition to the one that does best when all objectives are normalized and weighted equally. If the same system flight plan has the lowest value for multiple objectives, the second best system flight plan for one of the objectives is chosen until all five system flight plans are unique. The threshold for randomly selecting a parent from the entire population, p^{Sel}, is set to 0.8. This means that selecting parents from the entire population happens on average for 20% of the offspring in each iteration, which we consider to be a reasonable degree of randomness.

Table D.5: Overview of the values for the solution method parameters used in HEA$^{\mathrm{F}}$ and HEA$^{\mathrm{T}}$. The values above the horizontal line in the table refer to ACO parameters, while the parameters below the horizontal line refer to NSGA-III parameters.

Parameter [unit]	**Description**	**HEA$^{\mathrm{F}}$**	**HEA$^{\mathrm{T}}$**
ε	Constant used to calculate the probability of selecting next vertices in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	1	1
μ_0	Initial pheromone value per edge in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	10	10
ρ	Pheromone evaporation rate in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	0.5	0.5
φ	Pheromone decay rate in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	0.7	0.7
ω	Constant used to calculate the probability of selecting next vertices in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	7	7
A^{Init}	Number of ants in ACO$^{\mathrm{Init}}$	$4 \cdot \lvert\mathcal{S}_u\rvert$	$4 \cdot \lvert\mathcal{S}_u\rvert$
A^{Mut}	Number of ants in ACO$^{\mathrm{Mut}}$	4	4
I^{Init}	Number of iterations in ACO$^{\mathrm{Init}}$	6	6
I^{Mut}	Number of iterations in ACO$^{\mathrm{Mut}}$	5	5
P^{Init}	Number of paths to return from ACO$^{\mathrm{Init}}$	10	10
P^{Mut}	Number of paths to return from ACO$^{\mathrm{Mut}}$	1	1
Q	Amount of pheromone carried by each ant in ACO$^{\mathrm{Mut}}$ and ACO$^{\mathrm{Init}}$	1	1
c^{Min} [%]	Percentage improvement threshold	2	1
d	Number of divisions along each objective axis for reference point generation	4	6
I^{Min}	Minimum number of iterations	10	5
M	Number of flight plans/system flight plans to return	10	5
N	Population size	10	30
p^{Mut}	Threshold for performing a mutating crossover	0.8	0.8
p^{Sel}	Threshold for randomly selecting a parent from entire parent population	0.8	0.8

In the "Description" column we have ACO$^{\mathrm{Mut}}$: ACO used for mutating crossover, ACO$^{\mathrm{Init}}$: ACO used for initial parent population generation.

NTNU
Norwegian University of
Science and Technology