POLITECNICO DI TORINO

Master of Science in Mechatronic Engineering

Master's Degree Thesis

Route Planning of Unmanned Aircraft Systems for Urban Air Mobility application

Supervisors

Prof. Giorgio GUGLIERI

Dr. Stefano PRIMATESTA

Candidate

Angela Giulia PICCOLO

A.Y. 2021-2022

Abstract

Nowadays, the vision of incorporating aviation into everyday life is progressing because of strong urbanization and the evolving need for different transportation systems in metropolitan areas. A possible solution, as most recently discussed among research groups, could be found in UAVs used as Air-taxis: these are small electric aircraft that would provide passengers a faster and more convenient means of transportation compared to ground one.

According to this need, this work aims to plan routes for an Air Taxi fleet that are safe, short, low-energy, and optimally organized and distributed. In the literature, this multiple node routes service combination problem is known as UAV Vehicle Routing Problem (VRP).

For this purpose, several scenarios with different constraints have been evaluated. First, a simple and ideal scenario minimizing the vehicles' routes in time and distance.

Furthermore, the complexity of the scenario is increased by introducing other constraints: (i) capacity constraints of the vehicles (Capacitated VRP); (ii) pickup and deliveries location demanded by customers (VRP with Pickups and Deliveries); and, (ii) time scheduling of the rides according to customers calls for the service (VRP with Time Windows). The evaluation of these constraints influences the resulting solutions.

Lastly, a more complex scenario is evaluated, minimizing the vehicles' routes in time and distance, as well as considering the operational ground risk to third parties. This last feature is included by exploiting a risk-aware path planning algorithm capable of computing safe urban routes.

Proposed scenarios are tested through simulations using meta-heuristic methods since the VRP considered is an NP-hard problem. In particular, a use case in the city of Turin is considered by simulating an Air Taxi application and adopting the proposed route planning strategy.

Table of Contents

List of Tables

List of Figures

Acronyms

UAM
: Urban Air Mobility

VTOL
: Vertical Take-Off and Landing air vehicles

UAV
: Unmanned Air Vehicles

UAS
: Unmanned Aerial Systems

ARMD
: Aeronautics Research Mission Directorate

ATC
: Air Traffic Control

UOE
: UAM Operating Environment

UTM
: UAS Traffic Management

UTFC
: UAS Traffic Flow Control

FAA
: Federal Aviation Administration

TSP

Travelling Salesman Problem

CVRP

Capacited Vehicle Routing Problem

VRPTW

Vehicle Routing Problem with Time Windows

VRPPD

Vehicle Routing Problem with Pick-up and Delivery

RRT*

Rapidly-exploring Random Tree star

LS

Local Search

GLS

Guided Local Search

Chapter 1

Introduction

Nowadays, as technology and social needs are merging and evolving, the vision of incorporating aviation into everyday life has moved from the realm of sci-fi to a realistic and near-term opportunity. This evolution is progressing because of a strong urbanization and evolving preferences transportation in metropolitan areas.

Recently, this trend brought innovative projects to be investigated in order to improve the viability of the biggest cities. In particular traffic congestion, mainly related to wheel transportation, is increasing the level of air pollution and car accidents into metropolitan areas. Another drawback addressed to traditional urban transportation is related to the high dependence to fossil fuels: expensive and geopolitical unsafe resources.

Implementing those projects to the urban traffic network would improve citizens daily routine and welfare. For this reason **Urban Air Mobility (UAM)**, an emerging aviation transportation system that strives to commute passenger or cargo by air using low-altitude aircraft, is being widely investigated[1].

UAM is one of the possible solutions to escape urban gridlock throughout **vertical take-off and landing air vehicles (VTOL)** to create a network of air transportation routes through and between dense urban areas.

In order to reduce air pollution and make the urban traffic safer, UAM is conceived as populated by electric-powered vehicles and pilot-less aircraft guided remotely or autonomously.

Nowadays, this kind of vehicles known as Unmanned Air Vehicles (UAV), within Unmanned Aerial Systems (UAS), also or colloquially, drones, are used in civilian applications such as recreation, traffic monitoring, disaster monitoring, fire detection, infrastructure inspection, mapping, forestry, and agriculture: the objective is to introduce them to UAM traffic network.

Indeed, one of the last emerging application scenario is the **air-taxi**. It would be an upcoming ride-sharing service that is expected to compete against ground traffic through aviation.

UAVs used for Air-taxi purposed are small electric aircraft that will provide passengers a faster and more convenient mode of transportation compared to current public transportation systems that are already in place. Indeed, international transportation companies like Airbus, Uber and Kitty Hawk are working on research and development on such kind of UAV application.

1.1 Thesis goal and structure

According to this need, this work aims to plan routes for an air-taxi fleet that are safe, short, low-energy, and optimally organized and distributed. In the literature, this multiple node routes service combination problem is known as UAV Vehicle Routing Problem (VRP).

A state-of-the-art presentation of everything related to UAM is provided in Chapter 2. The following primary UAV focal points have been emphasized in several studies conducted by companies such as Volocopter, Uber, Amazon, and Airbus, which carried out on the main focal points of UAV. These are the following:

1. Infrastructure required to best support an air-taxi service, such as the establishment of Vertistop and Vertiport;
2. Specifications needed to create a UAV and a description of the eVTOLs now available on the market;
3. Geographic division and regulatory framework that allow the use of such a service in accordance with European civil aviation laws;

The problem addressed in the thesis study has been explained in Chapter 3. The thesis situation was detailed following an introduction to VRP and the primary restrictions related to the problem generally. In particular, a use case in the city of Turin is considered by simulating an air-taxi application and adopting the proposed route planning strategy. In Turin have been chosen the main Vertistops there to establish a strong air-taxi network, with an eVTOL Volocopter 2x as the test vehicle.

In Chapter 5 several tests have been proposed, initially a simple and ideal scenario minimizing the vehicles' routes in time and distance.

Furthermore, the complexity of the scenario is increased by introducing other constraints: (i) capacity constraints of the vehicles (Capacitated VRP); (ii) pickup and deliveries location demanded by customers (VRP with Pickups and Deliveries);

and, (ii) time scheduling of the rides according to customers calls for the service (VRP with Time Windows). The evaluation of these constraints influences the resulting solutions.

Lastly, a more complex scenario is evaluated, minimizing the vehicles' routes in time and distance, as well as considering the operational ground risk to third parties. This last feature is included by exploiting a risk-aware path planning algorithm capable of computing safe urban routes.

Proposed scenarios are tested through simulations using meta-heuristic methods, explained in Chapter 4, since the VRP considered is an NP-hard problem.

Chapter 2

Urban Air Mobility Application: air-taxi State of Art

The idea of flying vehicles within urban areas started in the 1940s, with the advent of helicopters. Low technology maturation, several accidents, together with high noise levels and high operating costs, forced most of the operators in the US and Europe to cease their operations in late 1960s to mid-1970s.[2]. Since then, the service aimed only at a niche of wealthy people, especially for long-distance travel.

Improvements and new discoveries in power electronics, sensors and data analysis, combined with significant cost reductions, have opened up entirely new possibilities, for UAVs, as it has been told before. Indeed companies as Uber, Airbus, Zephyir and Airworks are planning to launch this new safe, efficient and sustainable urban on-demand mobility service to citizens.

According to NASA's Aeronautics Research Mission Directorate (ARMD) most recent studies, the economic potential of UAM's viability has been examined.

These surveys showed that there could be large markets for UAM, but there are many technical, societal and regulatory challenges that need to be solved before UAM becomes a scalable operational concept accessible to many citizens and businesses.

Even though UAM and the use of EVTOL for passenger transport potentially is evolutionary as type of transportation, it is necessary to evaluate some challenges which are not only the safety of the vehicle design and certifications that the vehicles must possess, but also how safe it can be considered during its operation while flying and carrying commuters.

These challenges are the so called “barriers", which include any gap between the broad UAM community's current capabilities and what is required for UAM. Figure 2.1 shows what are the main challenges to be faced in the implementation of a complete air-taxi service: these frameworks have been identified by ARMD.

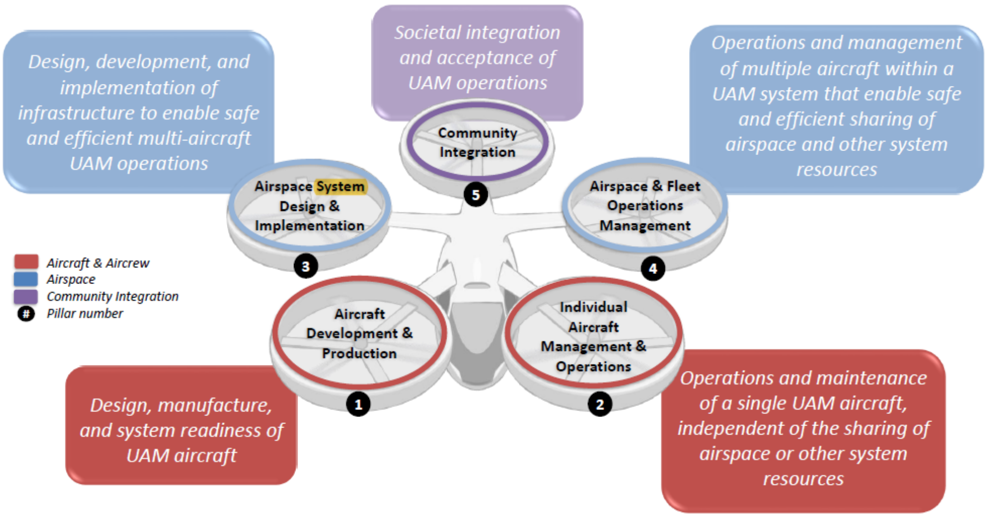


Figure 2.1: UAM organization frameworks [3]

Each of these frameworks summarized the main challenges to be faced in the implementation of air-taxi service with UAV, as explained above.

Aircraft Development and Production and Airspace System Design and Implementation pillars are related both with the design side of the framework.

The first deals with the design specifications of the vehicle, which must resist to adverse weather conditions, must control noise, and present a cutting-edge design that allows a minimum manufacturing cost. Before being placed on the market, all vehicles must possess the certifications released by competent agencies, as we will see in detail in Chapter 2.5.

The second challenge concerns the design of the airspace: construction of suitable infrastructures within the city indicated for the UAM; it also concerns the new division of the airspace, which must be modified, and the communication that must take place between the service providers operating in the new reality and the **Air Traffic Control (ATC)**.

On the right side of Figure 2.1 are highlighted instead the problems related to **Individual Vehicle Management & Operations** and **Airspace & Fleet Operations Management**. The first represents safe urban flight management and certification approval for each vehicle. The second framework involves adequately predicting and accounting for potentially unique environmental conditions at lower altitudes and near cities, managing higher volumes of air traffic. This pillar, which is treated in my study, includes challenges related to managing aircraft fleets planning a routing for UAV that enable operators to provide reliable UAM services to their customers.

The last challenge concerns **Community Integration**, whose focal point is to gain public acceptance according to the fact that UAM operations could beneficial to their communities and could be integrated into people's daily life.

Starting with infrastructure issues, the next few chapters will address the main challenges that UAM presents today.

2.1 Infrastructure requirements

Ground infrastructures, together with traffic management based on digital technology and telecommunications, play a very important role in support of UAM operations, representing constraints for the entire transportation network, as these affect the effectiveness of network operations.

UAM vehicles require a suitable take-off and landing infrastructure placed in strategic urban areas, such as hotels, offices, hospitals, schools, entertainment avenues and so on. These places are designed not only as stations, but as part of larger multi-purpose hubs for renewable energy, data, and public amenities: Figure 2.2 shows how are strategically distributed heliports in major USA cities.

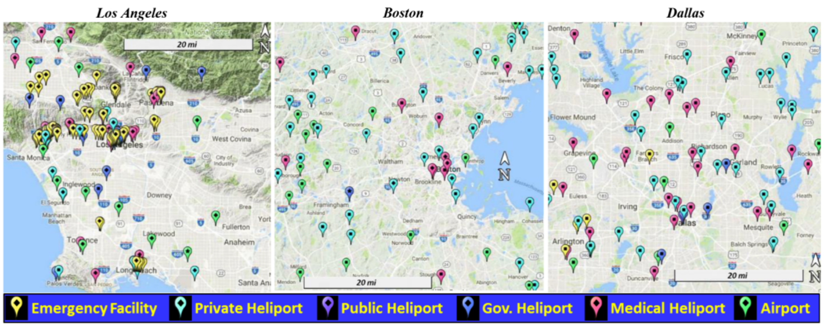


Figure 2.2: Existing heliport and airport infrastructure in Los Angeles, Boston, and Dallas. [4]

For these existing hubs there are not many changes to be made, but they could be integrated with bus station or railways.

According to Rajendran et al.[1], physical ground infrastructures dedicated for vehicles stand out in two types of assets:

1. **Vertistop** is a sophisticated rooftop helipad that handles a single air-taxi eVTOL at a time;
2. **Vertiport** is a platform used by multiple eVTOLs at a given time.

Vertiports are composed of an operational platform for rotorcraft manoeuvres and a connected building for associated technical equipment, such as charging infrastructure. There are various type of vertiports which presents two or more reception zone enabling higher capacity. Vertiports placement is choosen according to urban and environmental structure. Following some examples of vertiports settlements:

- rooftops;
- barges over water;
- inside highway cloverleafs;
- top of existing ground-transport infrastructure.

Various companies have developed and designed ideas of innovative transportation hubs. One of the first Vertiport network presented is made by **Volocopter**, UAM pioneer: in 2019 they published a handbook for the physical infrastructure supporting eVTOL aircraft operations highlighting direct collaborations with authorities, air operators, institutions, partners. In Figure 2.3 is shown one of the air-taxi Voloport project realised in Singapore.

Another company actively working and researching in the field, presenting innovative prototypes in the most important conferences is **Uber**.**UberAIR Skyports**, shown in the Figure 2.4,is expecting to begin tests for the service in Los Angeles and subsequently Dallas, Fort Worth and Frisco within 2023.

Figure 2.3: Voloport construction adaption on buildings @Volocopter [5]

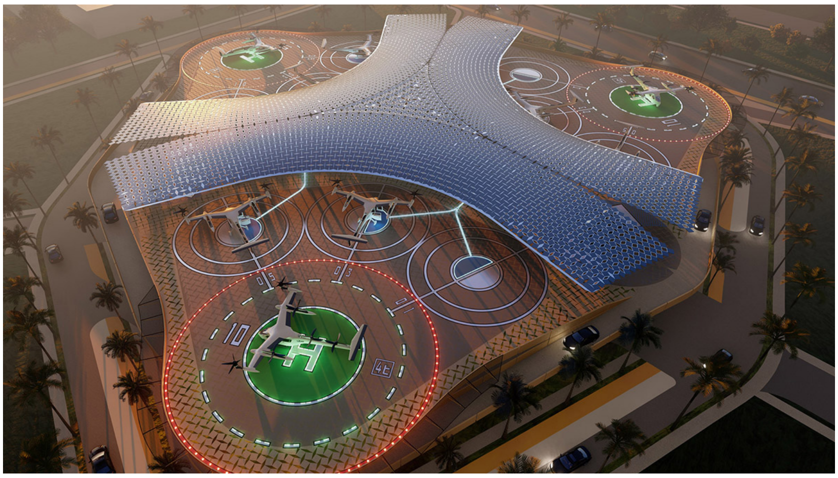


Figure 2.4: UberAIR Skyports @Uber

According to Vascik et. al.[6] work, focusing on infrastructure regulations for helicopters in re-size the derived minimum space requirements based on wingspans, lengths, and heights of different UAM designs that are in development.

Assuming that it is more efficient using existing heliports from the structural point of view, this choice should be a trade-off of strategic location and ease of access for the passengers.

2.2 UAV design

VTOL aircrafts development and design are key points to consider and evaluate, in order to successfully improve both urban mobility and spread the implementation of UAM. UAVs design and technical requirements are strongly influenced by optimal operating parameters and the end-use of the vehicles.

2.2.1 Technological requirements

There are many features for UAV design to be considered. Even though these are still in the experimental phase, some companies have developed prototypes that, at first glance, may have some features in common with current UAVs, mainly due to their appearance. Further than technical characteristics, even the approach to the surrounding environment and people acceptance must be evaluated in the design phase.

As shown in Figure 2.5, the diagram illustrates main design factors influencing the choice of a particular aircraft types.

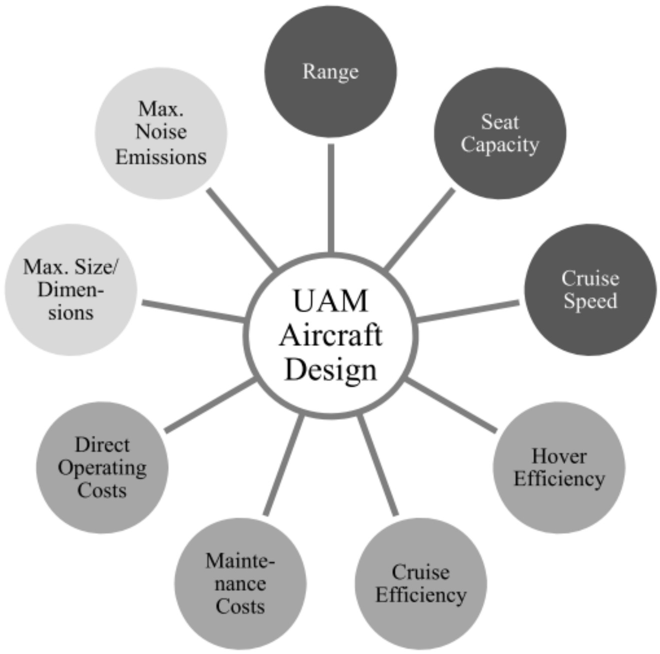


Figure 2.5: UAV main design factors [2]

These elements can be recollected into macro-groups:

- **Requirement**
  Mainly the different aircraft used in the air-taxi service can be grouped according to their performance given by the Range, Cruise control and Speed ;
- **External Boundary conditions**
  Acording to companies investing on UAM pilot-less, evaluating Direct Operating Costs, Maintenance Costs, Cruise Efficiency and Hover Efficiency is a crucial point. In fact, climb and landing maneuvers related to the overall missions shifts the optimum design point and may lead to different aircraft type ;
- **Public acceptance**
  air-taxi service clashes a lot with public acceptance and that is why it is necessary to work on noise emission reductions and on the maximum dimensions that these vehicles can assume;

In the recent years, several aircraft concepts have been designed according to the whole variety of these parameters. Each of this concepts has its pros and cons that would influence which model should be chosen according to end-user.

2.2.2 UAM VTOL models

According to Shamiyeh et al.[7], UAM VTOLs can be categorised into two classes of vehicles:

1. **Rotary Wing Cruise**

   This aircraft exploits the high rotational speed of its wings to generate a lift force able to raise from the ground. It naturally has very good hovering and VTOL characteristics. The rotor-craft group includes all types of multi-copter configurations as well as conventional helicopters. On the other hand, these aircrafts have big constraints regarding both cruise speed and flight efficiency.

   One reason of this lack of efficiency during flight operation can be related to the large rotor area: large footprints of the rotational wings can be avoided arranging them in a stacked configuration (see Figure 2.6a), at the expense of aerodynamic efficiency.

   More compact dimensions can be offered by the so-called lift-fan concepts which resembles an automobile (see Figure 2.6b). The encased rotors ensure higher level of safety and lower noise emission compared to other UAM concepts.

(a) **(b)**

Figure 2.6: Volocopter 2x @Volocopter (a) and Neva AirQuadOne @Neva Aerospace 2017 (b)

2. **Fixed Wing Cruise**

   This type consists of vehicles with wings ensembled with propellers enabling motion. These are significantly more efficient and faster in cruise flight compared to rotor-craft configurations. While this also increases the achievable ranges, the characteristics in hover and during VTOL flight are subject to compromises.

   Aircrafts with tilt propeller as in Figure 2.7b and tail float Figure 2.7a use the same propulsion system during all phases of flight, which requires a compromise in the design of the propulsion system.

   On the other hand, optimising dedicated propellers for cruise flight and switching off VTOL propellers during cruise flight can reduce efficiency losses. Sheathing or folding propellers that are not needed enables aerodynamically clean and thus very efficient cruise flight configurations. Necessary tilting mechanisms mean additional weight and increased system complexity.

2.3 Air Traffic Control Management

One of the most difficult constraint to be managed for UAM operations is the **Air Traffic Control (ATC)**. The main challenges in air traffic management are airspace integration, separation, contingency management, capacity, traffic flow, management, and scheduling

Nowadays the air traffic management system needs to be technologically advanced in order to become fully automated, easing its management from human air traffic control.

There are several studies carried on by both NASA and private industries

(a) (b)

Figure 2.7: Opener's BlackFly @OPENER (a) and The Joby S2 VTOL @Joby Aviation (b)

which have introduced two different philosophies of traffic control: the first one encourages that air traffic control should be centralized and technologically capable of accommodating aircraft at any level, while industries believes that vehicles should follow a well-defined route, especially if it is at a very high maturity level. **Maturity levels** are the developmental stages of UAMs, as defined by NASA. The maturity level we consider in this study is the fourth one: UML-4, which represents the initial phase of operations of eVTOLs for passenger traffic, the level of complexity is medium and the traffic volume is on the order of hundreds of concurrent operations.

Taking up the idea of having a well-defined route in the scenario of this project, in the Chapter 3 algorithms will solve UAV Routing Problem considering specifics routes through Turin, for air-taxi service purpose.

2.4 Airspace division

The introduction of an air-taxi service is closely linked to the need for a suitable and safe new airspace division. This must ensure proper cooperation between the various control systems now in place for the movement of public aircraft and private services.

For this reason, NASA has defined the so-called **UAS Traffic Management (UTM)**, a project designed to allow small, unmanned drones to access low-altitude beyond line-of-sight (BVLOS) airspace with minimal impact on the existing aviation system.

According to Bauranov et al.[8], airspaces in which flight operations occur are

Figure 2.8: UAM Operating Environment (UOE) [3]

named as **UAM Operations Environment (UOE)** (See Figure 2.8), and their volume isp redominantly static, so could be represented by standard aeronautical charts.

Taking into consideration the map of Turin from D-Flight website, society of ENAC group, pursues the development and provision of services for the management of air traffic at low altitude of remotely piloted aircrafts.

In the figure, it is possible to observe how the common spaces have been defined and highlighted with different colors depending on the allowable flight envelope over certain areas.

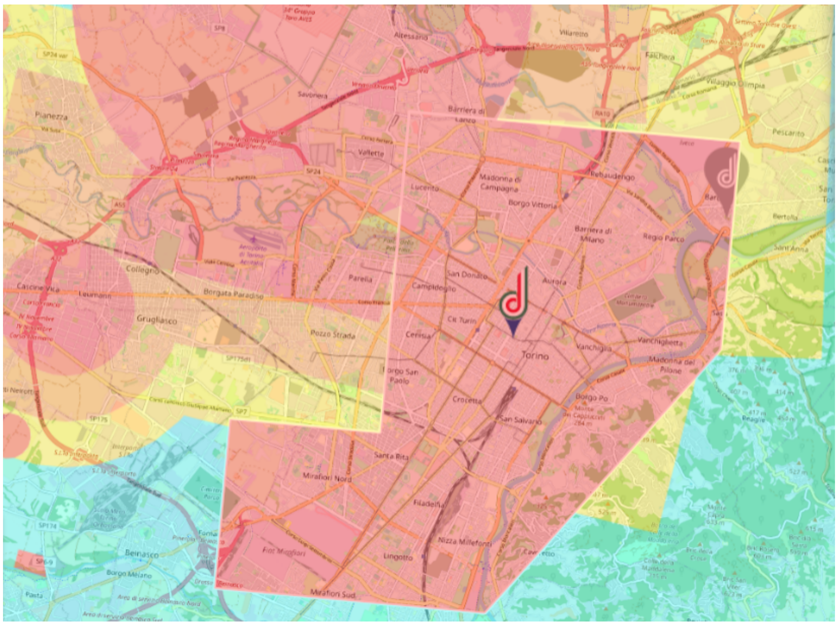


Figure 2.9: Flight restrictions over the city of Turin [**@D-Flight**]

The UOE is not a static environment. The restrictions highlighted on the map for certain areas of the city can be modified: could be applied some restrictions on legislation and temporary changes to airspace on the occasion of events. Restrictions may be changed for specific events, weather events, or heavy urban traffic.

The entity responsible for communicating real-time information about permitted flight zones is the **UAS Traffic Flow Control (UTFC)**. It controls density and throughput, monitors directional traffic flows, provides traffic information, identifies unauthorized flights, and sends safety advisories.

According to the Sunil et al., authors of the Metropolis project[9], four different types of urban airspace for drones have been proposed UAVs: full mix, layers, zones, and tubes. The idea of dividing the UOE into layers or not creating any ATC separation between UAMs and other operators has been shared by NASA Federal Aviation Administration (FAA).

Regardless of the space subdivision chosen, relationships between UAM, UTM, and ATM operations within different airspace classes must be considered to ensure the safety of passengers and urban populations.

Information exchange between different agencies is handled digitally and without human intervention: with continuous correspondence between the two route management agencies within the UOE.

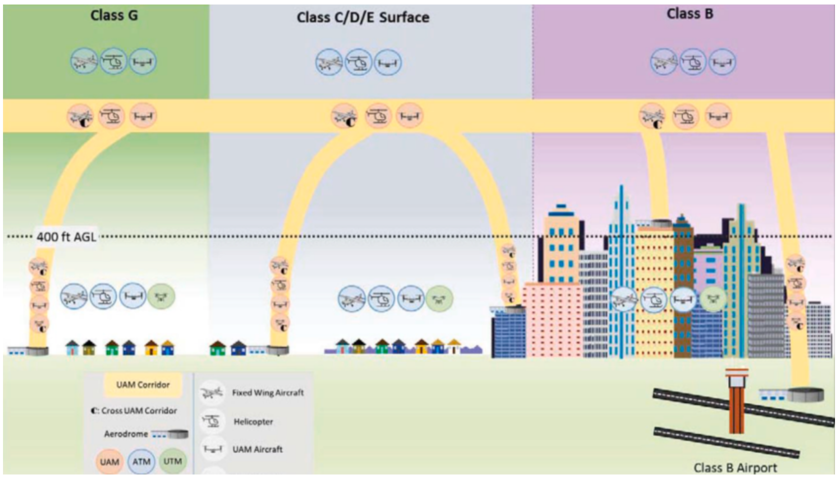


Figure 2.10: UAM, UTM and ATM Operating Environments
[8]

The private operator of each Vertiport is responsible for the so-called 'Vertiports operations', to allow UAVs to land and take off. It must apply for special permits to creates these corridors that intersect the space managed by the UOE. These are entities such as FAA and ENAC that receive real-time information from UTC, ATC, etc. to better manage space and create the best centralized and safe traffic management.

2.5 Regulatory Framework in Europe

An appropriate regulatory framework and legislation are the pillars of a proper UAV management. In the U.S., FAA is developing an Operational Concepts Map to regulate air transport within UOE, in collaboration with NASA.

However, if we look at the regulatory framework at the European and Italian level, we can observe how EASA (European Union Aviation Safety Agency) has recently issued, on 21/04/2021, 3 implementing regulations that initiate a radical change in the airspace structure. New provisions will be applicable as of 26 January 2023.[10]

These policy packages regulate **U-Space (Urban Air Space)**, a new concept of space. U-Space is a set of new services relying on a high level of digitization and automation of functions and specific procedures to support safe, efficient and secure access to airspace for a large number of drones. This is the European system being developed to manage drone traffic, comparable with the UOE defined by FAA.

The space thus designated includes operational boundaries defined by national bodies, in operational terms and also in terms of access requirements for UAS operators and the tools they use. The project proposes the implementation of 4 service packages to support the EU aviation strategy and regulatory framework for drones:

1. The first phase provides basic services such as identity registration (ID) and static geofencing to identify drones and inform operators about restricted areas.

2. The second phase connects drones to air traffic control and manned aviation. U-space provides initial services for managing drone operations, including flight planning, flight permit tracking, and connectivity to traditional air traffic control.

3. Advanced U-space services that support more complex operations in densely populated areas, such as conflict detection support, automatic detection and avoidance capabilities.

4. Comprehensive U-space services that provide a very high level of automation, connectivity and digitization for both the drone and the U-space system.

The application of U-Space to passenger transport is still in the study phase. During 2021 , the project CORUS-XUAM, organized by EASA SESAR division and menages by Eurocontrol, has made great progress in updating the operational concept of U-Space by adding aspects related to UAM.

At the Italian level, in 2019 the President of ENAC signed, with the Minister for Technological Innovation and Digitization, a memorandum of understanding for the launch of the national Urban Air Mobility (UAM) project "Innovation e-Mobility".

During the last years, a working group have been formed to identify AAM applications, including Air-taxi applications, find a solution for gaps and challenges to be overcome for the implementation of the selected business applications in the country and develop a clear Roadmap to fill the identified gaps and achieve the expected operational scenarios.

It has been estimated that by 2030 the achievement at Italian level of the objectives foreseen by the AML3 : a scenario involving a large scale commercial services , in a low density of UAVs and medium complexity operations within dedicated corridors.

Chapter 3

Urban Route Planning

Urban routes describe a transport network that needs to be planned and optimized, by considering that vehicles with different attributes and characteristics have to move from an origin to a destination to accomplish a specific task.

Several variables could affect the performance of the whole network, particularly in terms of the cost of operation. Indeed, the problem objective function, as it has been observed in Chapter 3.4, it has been modified by several constraints. The goal of this work is to optimize and develop the most efficient routes in terms of particular resources such as risk, time, distance, energy, or cost.

In this chapter we will look through UAV problem affected the study case of my work, giving a theoretical and mathematical formulation of problem description.

It has been examined how the UAV issue influenced the study-case of the work in this chapter, providing both a theoretical and mathematical definition.

3.1 UAV Routing Problem

Unmanned Aerial Vehicle routing is evolving from an emerging topic to a growing area of research. According to Rojas et al. [11], flexible 3D use of airspace by UAVs could potentially solve issues strictly related to urban mobility and logistics by reshaping the way these aspects are conceived. Several papers, such as , in this research field.

At the current status, despite of the increasing attention to UAVs and the maturity of this field, may be found in comprehensive overview of **UAV Routing Problem (UAVRP)** characteristics and the methods used to solve it, applicable to UAVRP for air-taxi service.

VRP problem has been introduced by Dantzing and Ramser in 1959; it generalises the well-known **Travelling Salesman Problem**, which considers a single vehicle visiting multiple customer locations before returning to the depot. The objective function is to minimize the total travel time or vehicle distance.

VRP is a multiple node service combination problem since it can generate multiple routes to pass through all customer locations. It is the basis of all routing literature and is the starting point to analise UAVRP.

As introduced in Adbelhafiz et al. [12], VRP goal is to minimized the total cost of operations and to optimize the routes to be taken by one or multiple vehicles, while in carrying passengers from a single point of origin, a depot, to a final destination identified on a map.

Generally, VRP is a static and deterministic spatial problem: since in this work has been considered some instances of it, as shown in the following chapters, to determine the optimal solution of UAVRP becomes a NP-hard problem. The last will be addressed in more detail in Chapter 4.

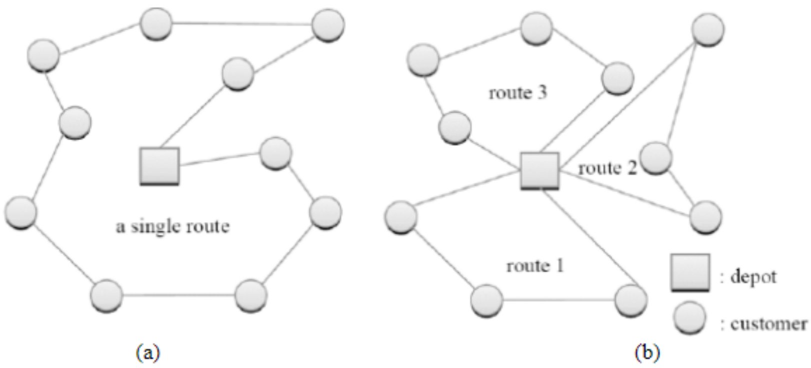


Figure 3.1: Different between: (a) traveling salesman problem (TSP) and (b) vehicle route problem (VRP)

[13]

In subsections below some strategical points on Turin map have been identified, considering the large amount of passengers that could benefit this service and possible infrastructure suitable for eVTOL, in particular focusing on air-taxi VRPs, temporal, capacities and pickups and deliveries aspects of routing problems.

3.1.1 VRP variants for air-taxi service

All the existing UAV's routing problems published during the last few years are either extended variants of the Vehicle Routing Problem. According to Faied et. al. [12], different constraints influencing VRP have been identified. More specifically for this study, it is possible to pick up the following instances, as more influent for UAM in an air-taxi service scenario:

- **Capacited Vehicle Routing Problem (CVRP)**
  CVRP is a VRP in which vehicles with limited carrying capacity need to pick up passengers from a depot to various location.
- **Vehicle Routing Problem with Time Windows (VRPTW)**
  VRPTW involve scheduling visits to customers who are only available during specific time windows. It is a restriction associated with each customer, defining an interval wherein the customer has to be supplied.
- **Vehicle Routing Problem with Pick-up and Delivery (VRPPD)**
  VRPPD is a VRP in which each eVTOL picks up passenger at various locations and drops them off at others.

3.1.2 VRP Problem Description

The city of Turin is considered as location of this case study for UAV routing problem analysis carried on this project.

Sixteen locations have been chosen within Turin: these points have been chosen as could be considered as the most important and strategic sites for creating an efficient network of air-taxis, where new *Vertiports* and *Vertistops* could be built.

The locations have been selected using *Geojson* tool, expressing them in geographic coordinates. One of them is fixed as *depot*, i.e. the stationing and charging station of transportation network selected: starting and ending point of each route. This choice fell on *Torino Porta Nuova* station due to the space available for a future construction of an eVTOL station for air-taxis service just in front of the main entrance.

As shown in Table 3.1, all the selected places have been list, together with their relative coordinates expressed in latitude and longitude. These stops also represents places where passengers are waiting for an air-taxi or are intending on going.

Vertistop	**Latitude**	**Longitude**
Torino Porta Nuova	45.06244	7.67868
Piazza Castello	45.07139	7.68504
Piazza Vittorio Veneto	45.06525	7.69503
Porta Palazzo	45.07743	7.68290
Stazione Torino Dora	45.09090	7.67700
Piazza Valdo Fusi	45.06400	7.68699
Italia 61	45.02876	7.66433
Stazione Torino Lingotto	45.02660	7.65680
Piazza XVIII Dicembre	45.07393	7.66821
Shopping Village Le Gru	45.05495	7.61328
Piazza Robilant	45.06129	7.64511
Parco Cavalieri di Vittorio Veneto	45.04668	7.65414
Politecnico di Torino	45.06252	7.66235
Ospedale Mauriziano	45.05191	7.66567
Ospedale Molinette	45.04143	7.67260
Parco Ruffini	45.05745	7.63935
Parco della Pellerina	45.08511	7.63935

Table 3.1: Depot and Vertistops geographical coordinates

In Figure 3.2, a map containing *Depot* and *Places* chosen in Turin city is shown . The framework has been taken from *Geojson.io* tool.

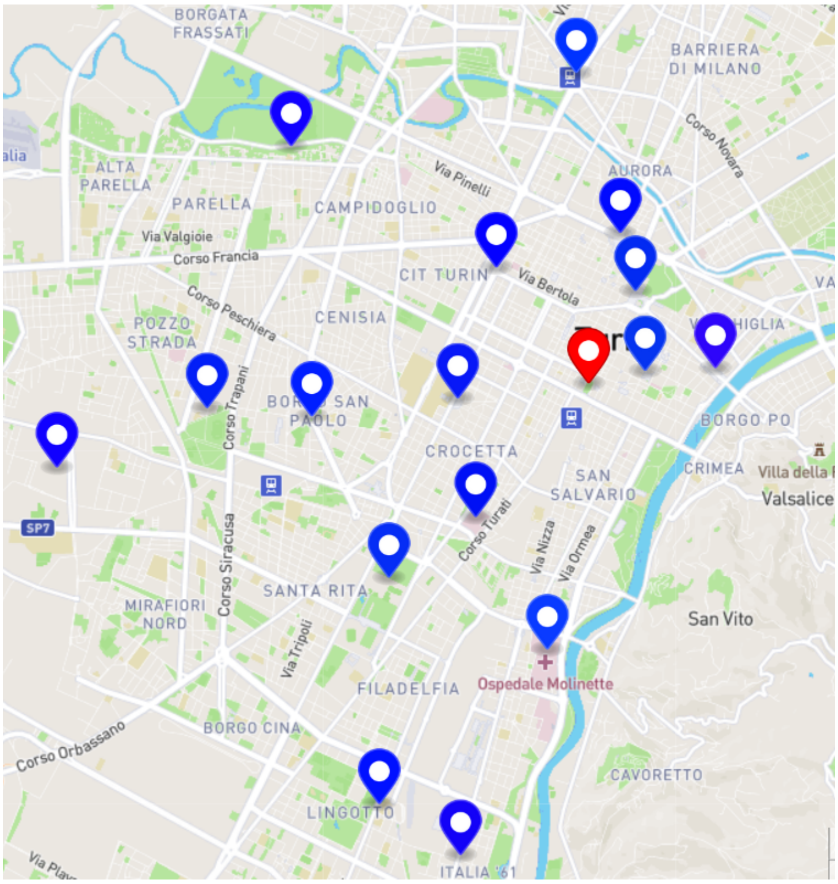


Figure 3.2: Turin map view of Depot (red pointer) and Vertistops (blue pointer)

The UAV considered to fulfill the service is **Volocopter 2x**. This is a *German two-seat, optionally-piloted, multirotor electric helicopter. The personal air vehicle was designed and produced by Volocopter GmbH of Bruchsal, and first introduced at the AERO Friedrichshafen airshow in 2017. The aircraft is sold complete and ready-to-fly* [14].

Since the eVTOL vehicle is optionally-piloted, it is necessary to highlight that in this work it will be unmanned guided. Its most important features have been discussed in the Chapter 2.2.2.

The optimization of routing problem discussed have been formulated as a Vehicle Routing Problem with **CVRP**, **VRPPD** and **VRPTW**.

Initially, the scenario, placed in Turin, considered for the various routing tests, discussed in Chapter 4, is set with no constraints on passengers but the vehicles itself. Subsequently, to create a realistic framework, the problem has been time to time complicated.

Firstly adding the drone passenger *Capacity* constraints, set to 2 passengers, as defined in the Volocopter specifications, and defining *Pickup and Deliveries* locations. Thus implies that the problem solver must consider each passenger's requests to go from the call point to a defined destination.

Next, *Time Window* costraints have been assigned for each passenger, which has to be respected into the limited time indicated, to satisfied transportation service.

Furthermore, the optimal solution route could include stopping at one location before the ending of commuting, landing and take off in order to drop one of the passenger, always according the maximum transportable weight fixed at 160 kg.

Adding consideration to the description problem, the Distance in meters from one destination to the other points, have been considered as the *cost* of each route. In the final test, instead, the *risk-based distance cost* of the different routes have been calculated taking into account a risk-map of Turin City.

The objective of the problem is to minimize the overall time of all the routes vehicles in time and distance factor, and the number of vehicles. This is a big deal since implies energy saving and minimization of fuel consumption.

3.2 Mathematical formulation

The aim of an optimization algorithm is to find the best solution among a wide range of different possible solutions.

Commonly to all optimization problems, two main elements affect the resolution of the problem:

- **The objective** quantity that occurs to be optimized. In this work, the objective have to minimized the cost expressed in times and distances. In order to set it up, a function must be defined which calculates the value of the objective for any possible solution. This is the so called **objective function**.

- **The constraints**, which are restrictions on the range of possible solutions, accordingly to specific assumptions made for the problem. A feasible solution is one that satisfies all the given constraints for the problem, without necessarily being optimal. The first step in solving an optimization problem is identifying the objective and constraints.

According to the study of S.C. Ho et al. [15], it has been possible to define the problem in a mathematical form. Making some assumptions and defining the following parameters, it is possible to express the problem's objective function with its constraints.

MODAL PARAMETERS

1. **Customers**: giving a set of customers $C = 1{,}2, ...n$, waiting in n different locations. Using 0 to indicated where depot is located, the set of all virtual field considered in the problem is $N = C \cup 0$; every costumers $i \in C$ has a demand $w_i > 0$

2. **Vehicles**: V indicates the fleet of vehicles, k with constant capacity m;

3. **Time Windows**: to each costumers correspond an interval $[a_i, b_i]$ where a_i and b_i represent the lower and upper bounder time limits where each vehicle may arrived to providing service, b_i, after that time the service cannot be satisfied anymore; additionally, the depot contains a time window $[a_i, b_i]$

DECISION VARIABLES, which characterized the problem's object function.

It is notice that for each arc (i, j), where $(i, j) \in N$, $i \neq j$, with only an exception of $i = j = 0$ which means vehicle is driving an empty route, is associated a cost of traveling $d_{i,j}$, distance in meters and a travel time $t_{i,j}$ expressed in seconds.

- $x_{ijk} = \begin{cases} 1\ if\ vehicle\ k\ travels\ directly\ from\ customer\ i\ to\ j \\ \qquad 0\ otherwise \end{cases}$

- $s_{i,k}$, specifies the time for the k^{th} tour to reach the i^{th} costumers site, where $s_{0k} = a_0$ for all k;

- $f_{i,k}$ is defined for each customer i and each vehicle k and denotes the fraction of demand of customer i delivered by the vehicle k.

OBJECTIVE FUNCTION

The objective function (3.1) below, minimized the total travel cost in time and distance:

$$min\ z(x) = \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} d_{ij} x_{ijk} \tag{3.1}$$

CONSTRAINTS

$$\sum_{j\in N} x_{0jk} = 1\forall k \in V \tag{3.2}$$

$$\sum_{i\in N} x_{ihk} - \sum_{j\in N} x_{hjk} = 0 \ \forall h \in C,\ \forall k \in V \tag{3.3}$$

$$\sum_{k\in V} f_{ik} = 1 \ \forall i \in C \tag{3.4}$$

$$\sum_{i\in C} w_i f_{ik} \leq m \ \forall k \in V \tag{3.5}$$

$$s_{ik} + t_{ij} - K_{ij}(1 - x_{ijk}) \leq s_{jk} \ \ \forall j \in C, \forall i \in N, \forall k \in V \tag{3.6}$$

$$a_i \leq s_{ik} \leq b_i \ \forall i \in N, \forall k \in V \tag{3.7}$$

$$s_{ik} + t_{i0} - K_{i0}(1 - x_{i0k}) \leq b_0 \ \ \forall i \in C, \forall k \in V \tag{3.8}$$

$$x_{ijk}(s_{ik} + t_{ij} - s_{jk}) \leqslant 0 \ \ \forall k \in V \ and \ (i,j) \in N \tag{3.9}$$

$$s_{ik} + t_{i,n+1} - s_{n+1,k} \leqslant 0 \ \ \forall k \in V andi \in C \tag{3.10}$$

$$f_{ik} \geq 0 \ \forall i \in C, \forall k \in V \tag{3.11}$$

$$s_{0k} = a_0 \ \forall k \in V \tag{3.12}$$

$$x_{iik} = 0 \ \forall i \in C, \forall k \in V \tag{3.13}$$

$$x_{ijk} \in 0{,}1 \ \forall i,j \in N, \forall k \in V \tag{3.14}$$

State constraints (3.2) and (3.3) establish that each costumer request is served once and only once, vehicle leaves the depot and after having satisfied passengers request, it returns to depot.

(3.4) state ensure that all passenger requests are met and (3.5) that the maximum loading capacity of C for all tours is not exceeded.

In order to respect the Time Window constraints applied for each costumers, it has been considered the (3.6) state inequalities defining the constant K_{ij}. A vehicle cannot arrive at customer j before $s_{ik} + t_{ij}$, so the constant has been defined as $K_{ij} = b_i + t_{ij} - aj$.

Constraints set (3.7) makes sure that all customers are serviced within time window limit and set (3.8) that vehicle return to the depot before the depot's time window closes.

Set of equations (3.9) and (3.10) imposes that pickup of passengers must happened before deliveries each of them.

3.3 Safe Route Planning

In order to find a probable UAV routing for air-taxi service in the city of Turin, to be as realistic as possible, it is necessary to evaluate the risk-based distance cost, expressed in distance meters, of each routes, which is not only given by aerial geographic distance. This is an extremely crucial point of the thesis work, if one wish to be able to trust the findings obtained by UAVRP solved of the last test in Chapter 5.4.

The risk-based distance cost is the length in meters of the minimum risk path connecting two point in the risk-map. Specifically, the risk-map is computed evaluating different factors, such as risk level, the no-fly zones and the presence of obstacles at the flight altitude, and subsequently, the risk-map is used to compute the minimum risk path using a Risk-Aware **Path Planning** algorithm.

A Risk-Aware Path Planning is an optimal path planning algorithm which minimized the overall risk to the population according to risk-map.

As it is proposed in the study by Primatesta et. al. [16], inputs of risk-map depends on environment characteristics, such as the *population density* and the *tridimensional model* of the urban area. These last influence the map layers.

The risk-map is a two-dimensional map divided into cell, where each of them represents a risk value associated. The last is assimilates to a matrix *R(x,y)*, in which the variables x and y are expressed in geographic coordinates, and denote the chosen points on the map in fig.3.2. The risk of each cell is computed evaluating the area involved in a possible crash of the UAV on the ground, as shown in Figure 3.3.

In this work, the risk value considered for defined the risk-map, has been defined as UAV poses to the general public by expressing frequency of causalities as a function of flight hour (h^{-1}), considering a possible ground impact accidents with a ballistic descent.

Indeed, the risk matrix according to its variables, assumes different risk values. Looking at the generated risk-map of the city of Turin, it is possible to distinguish different colors: in descending order from red, orange, yellow, and green, blue and purple risk values are indicated.

The map display has been done by means of the framework ROS and it has been visualized by the software **Rviz**.

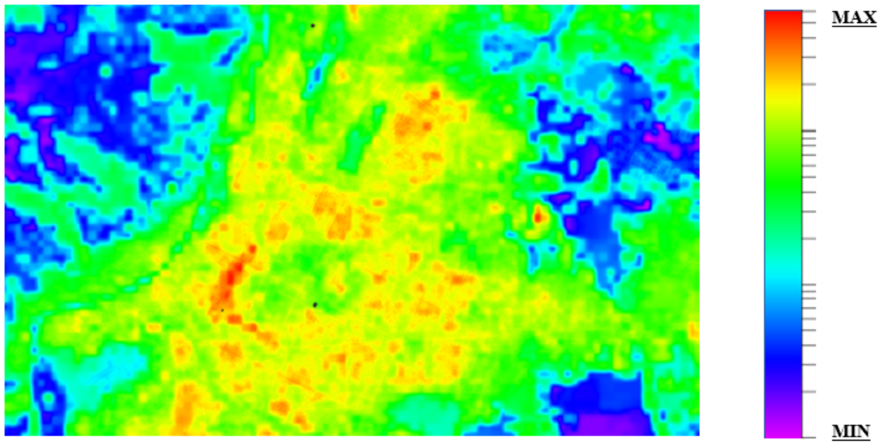


Figure 3.3: Turin risk-map with logarithmic scale of risk

Layers take into account for risk-map definition of Turin city are: [16]

- **Population density layer**: determines the population density and distribution on the map; this affects the likelihood that a person on the ground may be hit by a vehicle. It is a 2D location based map and each cell contains the population density value expressed in people/m2. In Turin city the average population density is 6939 people/km^2;
- **Obstacles layer**: defines the height of both fixed, buildings, and semi-static obstacles, construction sites, on the ground;
- **Sheltering layer**: It is a positive number that indicates the level of protection offered to people in each map cell by objects. These components lessen the kinetic energy during impact and, thus, the likelihood of failure. The matrix S defines a location-based map, and each of its elements, S(x, y), corresponds to a grid cell:

Area	Sheltering Factor
No obstacles	0
Sparse trees	2.5
Veichles and low building	5
High buildings	7.5
Industrial building	10

Table 3.2: Sheltering factor values

- **No-fly zone layer**: show on map the area in which UAV cannot overfly. Since UAVs are vehicles with special characteristics and different licenses used for different purposes, compared to those that are now on the market, the no-fly zones considered are different than those defined by National regulatory agencies;

According to Primatesta et al. work [17], the computation of risk assessment description have been done considering a probabilistic approach:

$$P_{casualty}(x, y) = P_{event} \cdot P_{impact}(x, y) \cdot P_{fatality}(x, y)$$

Thus means that to have a casuality in time $P_{casualty}(x, y)$ depends by:

- P_{event}: probability of losing control of the vehicle during an uncontrolled descent. In this work has been taken into account **ballistic descent** as descent type;
- P_{impact}: probability to impact a person when a UAV crashes on ground;
- $P_{fatality}$: probability of a fatal impact with a person;

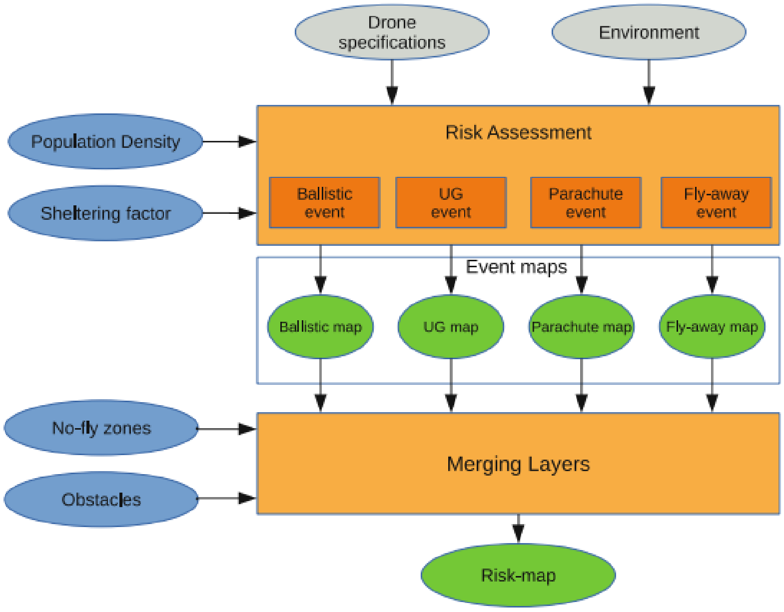


Figure 3.4: The main architecture of the risk-map generation [17]

Once risk-map of Turin city is generated, using the Grid map library [18], a C++ library interfaced with ROS (Robot Operating System), a risk-aware path planning is used to compute the minimum risk path in the map. Despite this, all the coordinates evaluated as Vertistop and the Depot, have been assigned as input, in order to obtained a network of paths.

The optimal path according to probability factors considered above, have been calculated applied an algorithm implemented in C++ based on the **Optimal Rapidly-exploring Random Tree star (RRT*)**, which has been proposed by Karaman and Frazzoli [19]: it is a sample-based algorithms that explore the search space with an incremental tree, connecting its to the branch with the minimum motion cost.

The algorithm minimizes a motion cost c_m that is computed incrementally in the exploration graph contructed by RRT*. The graph consists of nodes and edges and, then, each node has only one parent node and several children nodes. Defining $c_m(n_{i-1})$ as the motion cost of the parent node, r_{n_i} the risk function defined by the risk values in the risk-map, $\Delta t(n_{i-1}, n_i)$ is the flight time expressed in hour needed to cover two adjacent nodes, the **motion cost** is obtained applying the following equation:

$$c_m(n_i) = c_m(n_{i-1}) + \frac{r_{n_{i-1}} + r_{n_i}}{2} \Delta t(n_{i-1}, n_i) \tag{3.15}$$

The risk-aware path planning results has been obtained by imposing the following characteristics of the vehicle used for the simulations, Velocopter 2x:

Volocopter 2X specifications	
Velocity [m/s]	22
Capacity [-]	2
Maximum Payload [kg]	160
Maximum flight time [min]	43

Table 3.3: Volocopter 2X specifications

The paths have been implemented using the Open Motion Planning Library (OMPL) [20], an open source library specialized in sampling-based motion planning and it consists of many state-of-the-art algorithms. By initializing the geographic coordinates, showed in **Chapter 3.4**, and applying a *motionCost* function with the path planning algorithm RRT* (Figure 3.5) it will calculate the best path by analyzing the tree that *MotionCost* built, in order to minimize the risk-based distance cost.

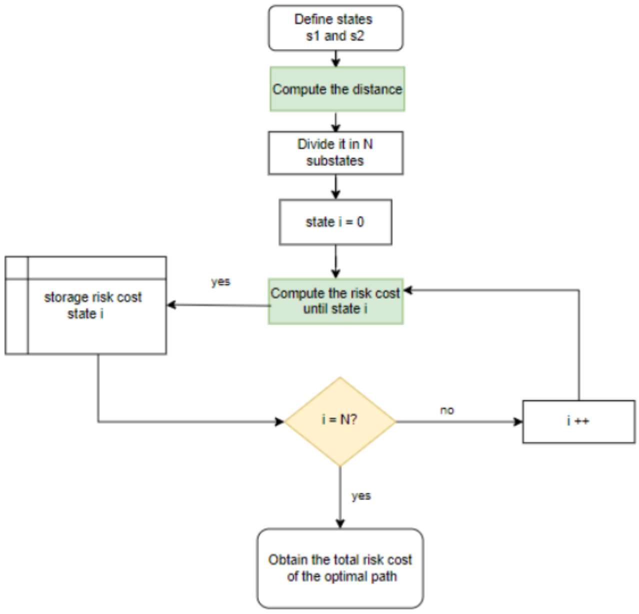
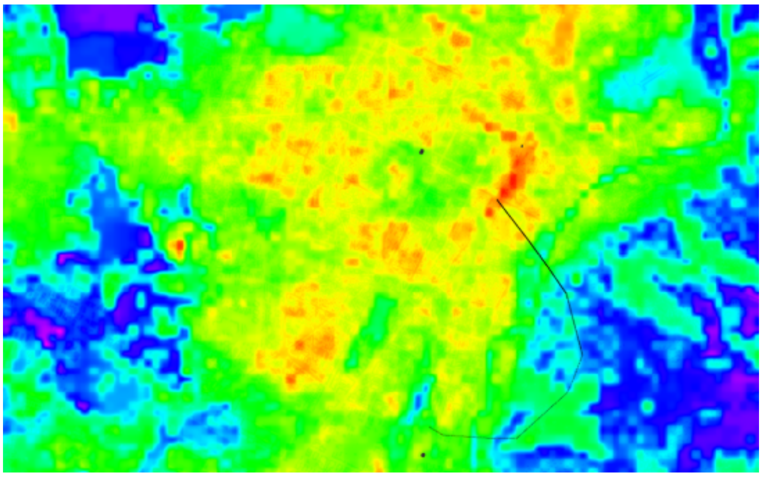


Figure 3.5: Minimization risk-based distance cost flowchart.
[**F. Fiorentino master thesis work**]

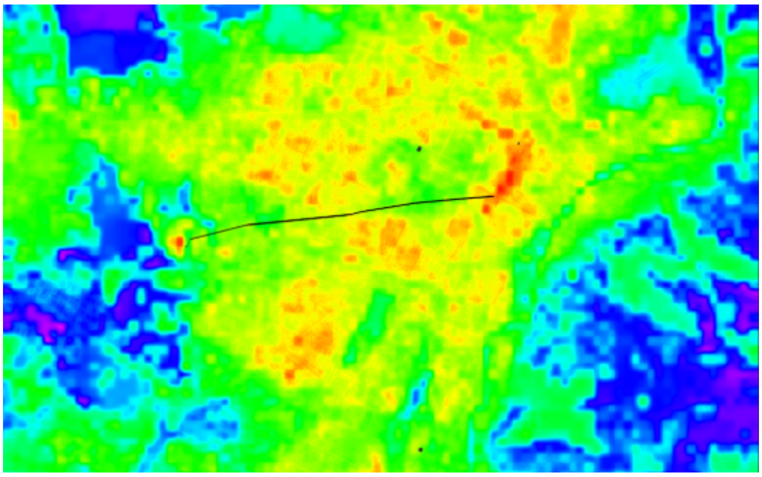
The optimal path obtained for the various routes considered will be used in the last test, the results of which are shown in Chapter 5.4.

The following Figures feature a several estimated scenarios: a black line represents the best path planning determined by minimizing the overall risk-based distance cost and flight time.

Indeed, it can be seen how the algorithm directs the vehicle to follow less risky paths, where there are areas colored purple, blue or green, rather than taking straight paths that minimize flight time but exponentially increase risk.



(a) Simulation of risk-awere path planning from Deopt 0 to Vertistop 7



(b) Simulation of risk-aware path planning from Depot 0 to Vertistop 9

Figure 3.6: Example of path planning simulation by minimizing the overall risk-based distance cost

Chapter 4

The Algorithm

As was already introduced in Chapter 3, the aim of the study is a routing problem which consists on finding the optimal or most feasible route for an air-taxi fleet operating in Turin urban area. This kind optimization problem is well known as Vehicle Routing Problem.

In this chapter it is explained the algorithm applied to UAVRP in order to obtained the optimal solution for the service provided.

4.1 Solution approach for UAVRP

VRP problems and their different applications, taken into account in Chapter 3, belong to a specific class of problem, whose algorithm chosen to find a feasible solution requires a polynomial time to solve it: this is the so called **Nondeterministic Polynomial (NP)** class problem. For this reason the problem to solve is considered NP-*hard*.

Variables such as weather conditions, random passenger demands and capacities, choosing the *Lowest flight Risk* zone, affect the global routing performance time and complexity: it results in a continuous process of collecting data, forming tours, and dispatching vehicles.

At the state of art several methods have been proposed to solve the original non-linear optimization problem: **Exact methods, Heuristic Algorithm or Metaheuristics Algorithm**. Although there has been significant progress in the development of accurate approaches, their application to NP-hard problems requires extremely long computation duration, so approximate approaches of resolution are so employed.

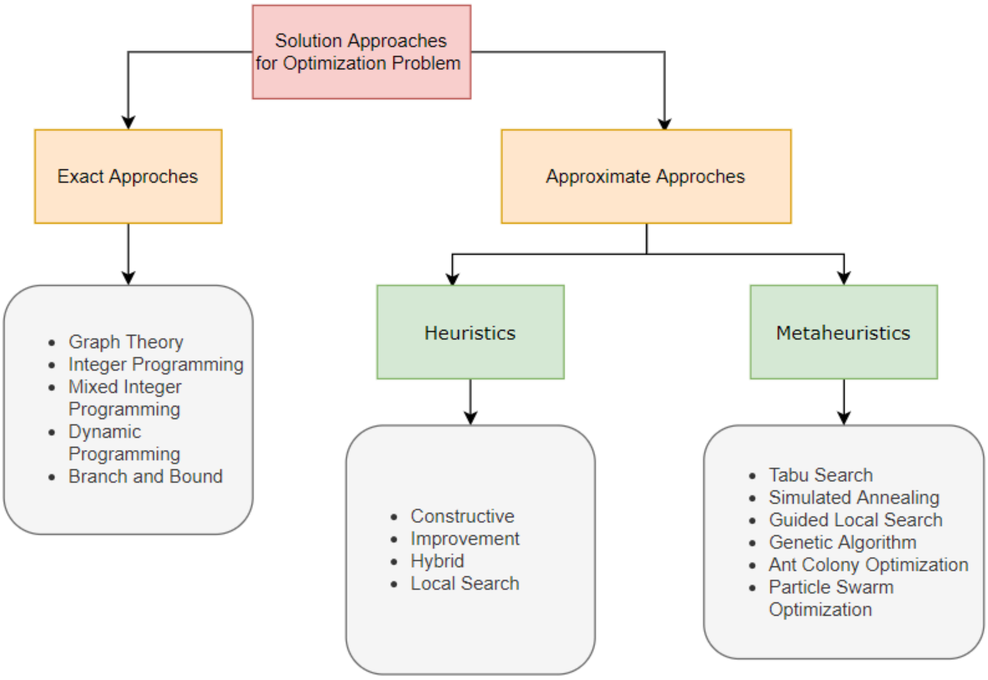


Figure 4.1: Taxonomy of Optimization Problems

Heuristic or Methauristic Algorithm provides a solution with computational time complexity for solving VRP which is exponentially proportional to the number of customer locations, (On^k), as *n=number of passengers and k=constant values.*

The major difference between exact and heuristic or metaheuristic methods is that the former guarantee the attainment of the optimal solution, while the latter seeks a good solution to the problem, not necessarily an optimal one, with shorter computational times.

One way to do this is to first create feasible routes for the vehicles to get a feasible solution (also referred to as, simply "solution"), and then improve upon them by injecting minor local changes like exchanging a few vertices. As described in Mohan et al. [21] work, these methods are collectively known as **Local Search** procedures.

The Local searches includes a general class of techniques, applicable to many different types of optimization problem. Of particular interest is the application of different techniques to the problem addressed in my work. In the following chapter will be addressed, explaining how different types of algorithms have been applied.

4.2 Local Search Algorithm

Local Search (LS) is the basis of many heuristic methods for combinatorial optimization, in particular it has been successfully employed for UAVRP applications: this success can be addressed to the use of intelligent exploration of the set of solutions.

It is an iterative method which can find good approximate solutions, based on trial and error concept.

Defining the combinatorial optimization problem in object , according to Voudouris et al. [22] work, as a pair (S, g), where S is the set of all feasible solutions and g is the objective function that maps each element s in S to a real number. The problem can be summarized as:

$$min\ \ g(s),\ \ s \in S \tag{4.1}$$

Thus, the goal is to find a solution that minimizes the objective function.

Adding contraints to the problem, the minimum value for the objective function is difficult to me: in these cases penalty terms may be added to $g(s)$.

Defining

$$N :\ \ S \to 2^S \tag{4.2}$$

$N(s)$ represents the *neighborhood* of s and contains all the solutions that can be reached from s by a single move.

A solution x is called **Local Minimum** of g if:

$$g(x) \leqslant g(y),\ \ \forall y \in N(x) \tag{4.3}$$

Concluding, Local search is a method of minimizing the cost function g by a series of iterative stages in which the current solution x is being replaced by a solution y such that:

$$g(y) < g(x),\ \ y \in N(x) \tag{4.4}$$

A fundamental local search algorithm starts with a random solution and when it reaches a *local minimum* further improvements are not possible.

Local search can be done in different ways, the one used by the solver employed in case study is the *first improvement* local search which accepts a better solution when it is found.

There are two major problems occurring once the LS method is applied: the first is the time taken, which increases exponentially with increasing neighborhoods; the second is that the search for the optimal solution stops at the local minimum, which could not be optimal, but good one instead. For this reason *Metaheuristics* will be involved in work, attempting to deal with this issues. These methods, when implemented to act on top of local search, try to escape the local search out of local optimal, *diversification*, and afterwards the search is continued in a new neighborhood, *intensification*.

Among all the major Metaheuristic methods, it is possible to find different procedures that could be applied: the one object of this work is **Guided Local Search**.

4.3 Guided Local Search

Guided Local Search (GLS) is conceived to escape a local optimum of the objective function, through modifying it: used to decide the direction of the comparison of Metaheuristics for a VRP search in the neighborhood of the current solution, including a set of *penalties*. Iterative calls are made to local search. Each time local search gets caught in a local minimum, the penalties are modified and local search is called again to minimize the modified cost function.

Before showing the Pseudocode of the GLS method, it is necessary to make some assumptions:

1. **Solution features**
   GLS solution is characterized by some *solution features*: any solution propriety that satisfies the non-trivial simple constraint qualifies as a solution feature.

$$I_i(s) = \begin{cases} 1 & solution\ s\ has\ propriety\ i \\ 0 & otherwise \end{cases} \quad s \in S \tag{4.5}$$

   For Vehicle Routing Problem, the features could be the Locations selected and the feature cost could be the Locations costs.

2. **Augmented cost function**
   By adding a set of penalty terms to the problem's cost function, restrictions on features are made possible. The new cost function is called *augmented cost function* and is defined as follow, where M is the number of features defined over solution, p_i the penality parameter corrisponding to feature f_i and *lambda* a parameter for controlling the strenght of constraints with respect to the actual solution cost.

$$h(s) = g(s) + \lambda \sum_{i=1}^{M} p_i I_i(s) \tag{4.6}$$

GLS iteratively uses local search passing it the augmented cost function for minimization and it simply modifies the penalty vector p given by $p = (p1, ..., pM)$ each time local search settles in a local minimum. Since GLS modifies the local minima's status under the augmented cost function using the penalty modification mechanism, the local minima found by local search when GLS is used may differ from the local minima found by local search when GLS is used with the original cost function of the problem.

3. **Penalty modifications**
When local search gets stuck in a *local minimum* s_*, the penalty modification process is what controls the augmented cost function, which are incremented by one for all features f_i that maximize the utility expression:

$$until(s_*, fi) = I_i(s_*)\frac{c_i}{1 + p_i} \tag{4.7}$$

In Figure 4.2 has been depicted the GLS algorithm applied in this work in pseudo-code.

```
procedure GuidedLocalSeach(S, g, λ, [I_1, ...,I_M], [c_1,...,c_M], M)
begin
    k ← 0;
    s_0 ← random or heuristically generated solution in S;
    for i ←1 until M do /* set all penalties to 0 */
        p_i ← 0;
    while StoppingCriterion do
    begin
        h ← g + λ * ∑p_i*I_i;
        s_k+1 ← LocalSearch(s_k, h);
        for i ←1 until M do
            util_i ← I_i(s_k+1) * c_i/ (1+p_i);
        for each i such that util_i is maximum do
            p_i ← p_i + 1;
        k ← k+1;
    end
    s* ← best solution found with respect to cost function g;
    return s*;
end
```

where S: search space, g: cost function, h: augmented cost function, λ: lambda parameter, I_i: indicator function for feature i, c_i: cost for feature i, M: number of features, p_i: penalty for feature i.

Figure 4.2: Guided Local Search in pseudocode
[22]

4.4 Google OR-Tools Implementation

In this thesis, metaheuristics algorithm have been implemented using an optimization software **Google OR-Tools**.

Google itself provides an environment to use and explore routing problems. The software exploits meta-heuristics in its libraries, able to solve optimization problem by exploring ranges of possible solution and to overcome local optima which would affect the final result of the optimization problem.

In order to solve the VRP problem proposed, applying Metaheuristic algorithm is important to set *Hypothesis* and *Constraints*.

Hypothesis chosen for the thesis case-study are: *number of vehicles*, *depot* and the *mutual distances (Distance Matrix)* among all the locations for the Vertistops.

Constraints are related to the characteristics itself of the problem: in this particular case air-taxi application. Taxis works with passengers, so a maximum load of passengers, the Pick-up and Drop-Off of the passengers and Time Windows constraints must hypotized.

Both **Contraints** and **Hypothesys**, contribute to the creation of a **Data Model**. These will be inputs for function which would be inserted into graph nodes, in order to define the combined cost including both distance and capacity for each node or arch.

Data model differs in the various cases addressed in the thesis, depending on the constraints considered. The following subsections explain all the various scenarios tested to which the methods mentioned above were applied.

4.4.1 UAV Routing Problem without constraints

The first running test has been done without considering any constraints for the UAVRP. The *Data model*, in this case, consists of *Distance Matrix*, *Number of vehicles* and depot definition, which in all the case will be associated to *Torino Porta Nuova* location.

The Distance Matrix represents a numerical **D(x,y)** matrix, where x and y are the coordinates of the depot and all the Vertistops taken into account for this work. Each element of the matrix is distances between network routes, expressed in meters.

In this case this Distance Matrix only considers point to point length, since risk assessment is not taken into account in this scenario.

The Matrix has been derived using function from *python tsp library.* By providing the coordinates of the chosen points, as an array to the *great circle distance matrix* function, it returns the Distance matrix.

Aiming to identify a **Routing Model**, it has been necessary to locate the Routing indices, which would represent Depot and Vertistops. Each distance between nodes is associated to a cost: *SetArcCostEvaluatorOfAllVehicles* method translate the distances expressed in meters into a dimension associated to their network map node. It has been realized through *pywrapcp* function, from *ortools constraints solver* library.

Subsequently, the only constraint applied to this test has been *Distance Dimension*, imposing 38 km as the maximum limit that each vehicle could travel, in line with the eVTOL's specifications.

According to problem definition, it has been possible to obtain an initial solution applying to the solver a method. The methods applied to the problem, and subsequently tested, were two: **Local Cheapest Insertion** and **Parallel Cheapest Insertion**.

The first one is a technique which iteratively build a solution by inserting each node at its cheapest position; the second method applied considers nodes in their order of creation. *Local Cheapest Insertion* is more faster than *Parallel Cheapest Insertion.* [23]

The results obtained through the first solution strategy can be considered optimal, without having to recur to any other more heuristic or metaheuristic method. Several tests have been conducted to observe the different results obtained, as will be shown in Chapter 5.

The test results obtained do not apply to my study case. It have been carried out to highlight how in an ideal study case without any constraints, an optimal solution could be derived through the application of Google OR Tools software routing methods.

4.4.2 UAV Routing Problem with Capacities and Pickup and Deliveries constraints

The second VRP formulation problem tested for air-taxi application study case is the Capacity VRP and VRP Pickup and Deliveries problems.

As the previous case, the scenario is based on a ideal map of Turin with no risk considered with the addition of new constraints **Pickup and Deliveries** and **Capacity**.

Constraints connected to passenger pickup and delivery have always been taken into account for every problem formulation that has been defined after this one. It is important to include for each vehicle a limitation to pick up and drop off passengers at every locations, in order solution for air-taxi VRP to be deemed a feasible option.

Constraints imposed to each vehicle before departure, might be thought as the beginning and the end of each passenger's journey. The problem is to assign routes for the vehicle, minimizing the cost of the journey. Each passenger is related to an interval in which there is indicated the starting point and the desired final destination of the journey.

	Pickup Locations	Delivery Locations
Passenger 1	1	6
Passenger 2	3	12
Passenger 3	5	7
Passenger 4	10	13
Passenger 5	14	16
Passenger 6	9	2
Passenger 7	4	15
Passenger 8	8	17
Passenger 9	11	18

Table 4.1: Pickup and Deliveries Constraints

In addition, the VRPPD problem must deal with **Capacity** constraints (**CVRP**). These refer to a vehicle's maximum load capacity, which is set to 2 passengers.

These constraints are added to the *Data Model* set. This was set by considering the distance between the various indices of the Distance Matrix and setting a Distance dimension, exactly as in the previous case. In addiction to the Distance callback, the solver in this test required a *Demand Callback* which returns the demand at each location, and a dimension for the capacity constraints. *Demand* is the passenger demand located in each position, which is intended to symbolize loading and unloading of passenger in each Vertistop.

Once all the data have been acquired, the solver set the method of **Parallel Cheapest Solution**, as *First Solution Strategy.* Since in this case the problem falls into the class of hard NPs, it is necessary to apply a metaheuristic algorithm to find an optimal solution or one that comes as close to it as possible.

The solver can escape a local minimum, a solution that is shorter than all nearby routes but is not the global minimum, by using a more sophisticated search technique called *Guided Local Search.* The solver then moves on from the local minimum and continues the search is **Metaheuristic**.

The metaheuristic algorithm applied in this case is The **Guided Local Search** builds up penalties during a search. It uses penalties to help local search algorithms escape from local minima and plateaus. [24]

The routes obtained will be shown in Chapter 5.2 .

4.4.3 UAV Routing Problem with Time Windows constraints

The following scenario belongs to a category of VRPs with **Time Windows** constraints (**VRPTW**). This problem has the same formulation as the problem discussed in the previous section, with the addition of time constraints.

It has been possible to associate the constraints applied to each passenger as a range of waiting time in which, after ride booking, the passenger waits its vehicle.

The time constraints have been also applied for passengers destinations, i.e., it represents the maximum ranges in which the drone is constrained to arrived. These are such targets to optimize passenger travel time within a city.

Time windows constraints have been set differently in the various tests to observe different behavior of the algorithm as we will see in the results in chapter 5.3. The final constraint was set to recreate a scenario as close to reality as possible, have been fixed to [0,1000s] for Vertistop and [0,0] for the Depot.

Plus time window, the *Capacities and Pickup and Delivers* constraints have been taken, also, into consideration.

The difference from previous tests is that in this case, the *Data Model* is the **Time Matrix**: this is the network of paths between locations that expresses the time it takes to go from one point to another. This is also assimilated into a matrix *R(x,y)*, where x and y are the distance expressed in seconds. This has been derived from the Distance Matrix, setting drone speed at 22 m/s.

Each element of the *Time Matrix* has been assigned as the cost between points, to the *Routing Model* with function *SetArcCostEvaluatorOfAllVehicles*. Then the constraints for the *Transport*, and the constraints for the *Depot* and each *Vertistops* have been defined. The waiting time for each passenger vehicle has been set to **2 min**; while the maximum flight time per vehicle to **43 min** to meet the specifications of the *Volocopter* company.

By setting the *First Heuristic Solution* it is used by solver as input for the metaheuristic algorithm applied. *Guided Local Search* reconfirm the best in this case, as it is possible to compare through the results obtained into Chapter 5.3.

Chapter 5

Results

In this Chapter a numerical analysis is performed to test the algorithms previously discussed.

The computational work has been carried out with a laptop HUAWEI Matebook D15 equipped with a quad core Intel ® Core ™i5-1135G7 2.4 GHz and 16 GB of RAM installed. The code have been developed in Microsoft Visual Studio 2022 using Python 3.9.

The vehicle chosen is a **Volocopter 2X**, according to this choice it has been set the following initialization parameters:

Volocopter 2X specifications	
Velocity [m/s]	22
Capacity [-]	2
Maximum Payload [kg]	160
Maximum flight time [min]	43
Rate of descent [m/s]	2.5
Altitude [m]	100
Noise level [dB]	87
Power supply	9 lithium-ion battery system for each of two motors
Propulsion	3 phase PM synchronous motor, brushless DC electric motor (BLDC)

Table 5.1: Volocopter 2X specifications

As mentioned above, for the first three tests, the experiments have been conducted by considering an ideal map of Turin, without taking into account risk-aware path planning evaluation. About this, the cost matrix of the VRP network corresponds to the distance, expressed in meters, between the various routes.

Then, for the last test through the RRT* algorithm, for each route a path planning have been evaluated, in order to consider a real scenario minimizing the risk of vehicle in flight. The cost matrix obtained is expressed in meters: it is important to highlight that each time the algorithm is compiled, it provides different values of the distance between the routes. Thus, the cost changes not for the direction, but because the algorithm is not deterministic. Therefore, to each calculation it provides a different, albeit similar solution.

Before showing the results, it is necessary to evaluate some assumptions. Since the algorithm associates each Vertistop with the start or the end point variable, in *Vertistop 4* and *Vertistop 13* besides being the final destination of some passengers, these are also starting points where other passengers are waiting to be transported.

Consequently, it was necessary to define *Vertistops 4 and 13* again, associating their coordinates with **Vertistops 17 and 18**, respectively, to make sure that the routing algorithm works properly.

5.1 Test 1 - air-taxi service in Turin: no constraints considered

The first running test has been done without considering any constraints for the UAVRP, selecting only the distance matrix as input.

The map considered is ideal, with no risk coefficient applied in any area.

The *Distance Matrix* values shows in 5.2.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	1112	1321	1698	3166	674	3911	4340	1519	5204	2640	2605	1282	1554	2385	3574	3986	3166	1554
1	1112	0	1039	692	2259	835	5011	5452	1352	5925	3333	3666	2036	2647	3471	4317	3898	2259	2647
2	1321	1039	0	1655	3183	646	4720	5243	2317	6522	3945	3818	2584	2742	3181	4892	4898	3183	2742
3	1698	692	1655	0	1567	1527	5605	6012	1218	6011	3468	4098	2313	3143	4083	4455	3524	1567	3143
4	3166	2259	3183	1567	0	3092	6980	7323	2009	6404	4136	5234	3358	4425	5511	5037	3025	0	4425
5	674	835	646	1527	3092	0	4303	4787	1843	5876	3303	3219	1942	2147	2752	4246	4416	3092	2147
6	3911	5011	4720	5605	6980	4303	0	638	5032	4956	3919	2147	3757	2576	1551	3994	6565	6980	2576
7	4340	5452	5243	6012	7323	4787	638	0	5338	4650	3965	2242	4017	2899	2064	3879	6648	7323	2899
8	1519	1352	2317	1218	2009	1843	5032	5338	0	4802	2294	3225	1349	2456	3630	3269	2583	2009	2456
9	5204	5925	6522	6011	6404	587636	4956	4650	4802	0	2597	3338	3945	4129	4896	1630	3929	6404	4129
10	2640	3331	3945	3468	4136	3303	3919	3965	2294	2597	0	1772	1360	1922	3088	990	2687	4136	1922
11	2605	3666	3818	4098	5234	3219	2147	2242	3225	3338	1772	0	1875	1076	1563	2001	4428	5234	1076
12	1282	2036	2584	2313	3358	1942	3757	4017	1349	3945	1360	1875	0	1208	2478	2317	3093	3358	1208
13	1554	2647	2742	3143	4425	2147	2576	2899	2456	4129	1922	1076	1208	0	1286	2583	4230	4425	0
14	2385	3471	3181	4083	5511	2752	1551	2064	3630	4896	3088	1563	2479	1286	0	3535	5514	5511	1286
15	3574	4317	4892	4455	5037	4246	3994	3879	3269	1630	990	2001	2317	2583	3535	0	3107	5037	2583
16	3986	3898	4898	3524	3025	4416	6565	6648	2583	3929	2687	4428	3093	4230	5514	3107	0	3025	4230
17	3166	2259	3183	1567	0	3092	6980	7323	2009	6404	4136	5234	3358	4425	5511	5037	3025	0	4425
18	1554	2647	2742	3143	4425	2147	2576	2899	2456	4129	1922	1076	1208	0	1286	2583	4230	4425	0

Table 5.2: Distance matrix of ideal cost [m]

Taking into account Time Matrix shown in Table 5.3, which will be applied in the third test, it returns the necessary time for each UAV to move from a point to another one through Turin, expressed in seconds. It has been calculated through a *Distance Matrix* (5.2) and assuming a UAV constant velocity expressed in m/s, according to the following formula:

$$t = \frac{s[m]}{v[m/s]}[s]$$

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	50	60	77	143	30	177	197	69	236	120	118	58	70	108	162	181	143	70
1	50	0	47	31	102	37	227	247	61	269	151	166	92	120	157	196	177	102	120
2	60	47	0	75	144	29	214	238	105	296	179	173	117	124	144	222	222	144	124
3	77	31	75	0	71	69	254	273	55	273	157	186	105	142	185	202	160	71	142
4	143	102	144	71	0	140	317	332	91	291	188	237	152	201	250	228	137	0	201
5	30	37	29	69	140	0	195	217	83	267	150	146	88	97	125	193	200	140	97
6	177	227	214	254	317	195	0	29	228	225	178	97	170	117	70	181	298	317	117
7	197	247	238	273	332	217	29	0	242	211	180	101	182	131	93	176	302	332	131
8	69	61	105	55	91	83	228	242	0	218	104	146	61	111	165	148	117	91	111
9	236	269	296	273	291	267	225	211	218	0	118	151	179	187	222	74	178	291	187
10	120	151	179	157	188	150	178	180	104	118	0	80	61	87	140	45	122	188	87
11	118	166	173	186	237	146	97	101	146	151	80	0	85	48	71	90	201	237	48
12	58	92	117	105	152	88	170	182	61	179	61	85	0	54	112	105	140	152	54
13	70	120	124	142	201	97	117	131	111	187	87	48	54	0	58	117	192	201	0
14	108	157	144	185	250	125	70	93	165	222	140	71	112	58	0	160	250	250	58
15	162	196	222	202	228	193	181	176	148	74	45	90	105	117	160	0	141	228	117
16	181	177	222	160	137	200	298	302	117	178	122	201	140	192	250	141	0	137	192
17	143	102	144	71	0	140	317	332	91	291	188	237	152	201	250	228	137	0	201
18	70	120	124	142	201	97	117	131	111	187	87	48	54	0	58	117	192	201	0

Table 5.3: Time Matrix considering ideal map of Turin [s]

Applying the algorithm discussed in **Chapter 4.4.1**, the route planning obtained is the following:

Vehicle 1	
Route	0 → 4 → 17 → 3 → 1 → 2 → 5 → 0
Distance [m]	7784
Time Required	5 min 51 s

Vehicle 2	
Route	0 → 18 → 13 → 11 → 7 → 6 → 14 → 0
Distance [m]	9446
Time Required	7 min 6 s

Vehicle 3	
Route	0 → 9 → 15 → 0
Distance [m]	10408
Time Required	7 min 52 s

Vehicle 4	
Route	0 → 12 → 10 → 16 → 8 → 0
Distance [m]	9431
Time Required	7 min 7 s

Vehicle 5	
Route	0 → 0
Distance [m]	0
Time Required	0 min 0 s

Vehicle 6	
Route	0 → 0
Distance [m]	0
Time Required	0 min 0 s

Table 5.4: Test 1.1 - routing results with no-constraints considered. First solution strategy applied Parallel Cheapest Insertion

Vehicles 5 and 6 have not been used, as can be seen from the results: mainly because the algorithm, when is seeking for optimal solution, sought to minimize the total time that drones were in flight, rather than the sum of the minimum times that each vehicle was in flight.

On this reason, even though using another vehicle for a route would save either time or distance, it was also considered as a potentially additional risk and energy cost.

Results	
Maximum of the route distances	37 km and 69mt
Total time of the routes	0 h 27 min and 56 s
eVTOLs used	4

Table 5.5: Overall results of test 1.1, with first solution strategy applied Parallel Cheapest Insertion

These results have been obtained by applying as first solution strategy, *Parallel Cheapest Insertion*.

In Fig. 5.1 below, routing results obtained have been graphed.

Several tests have been conducted to observe the different results obtained by setting the various methods as First Solution strategy, proposed by Google OR-Tools.

Most relevant tests have been showed in Table 5.6. The result has been obtains by applying **Local Cheapest Insertion**, which iteratively build a solution by inserting each node at its cheapest position.

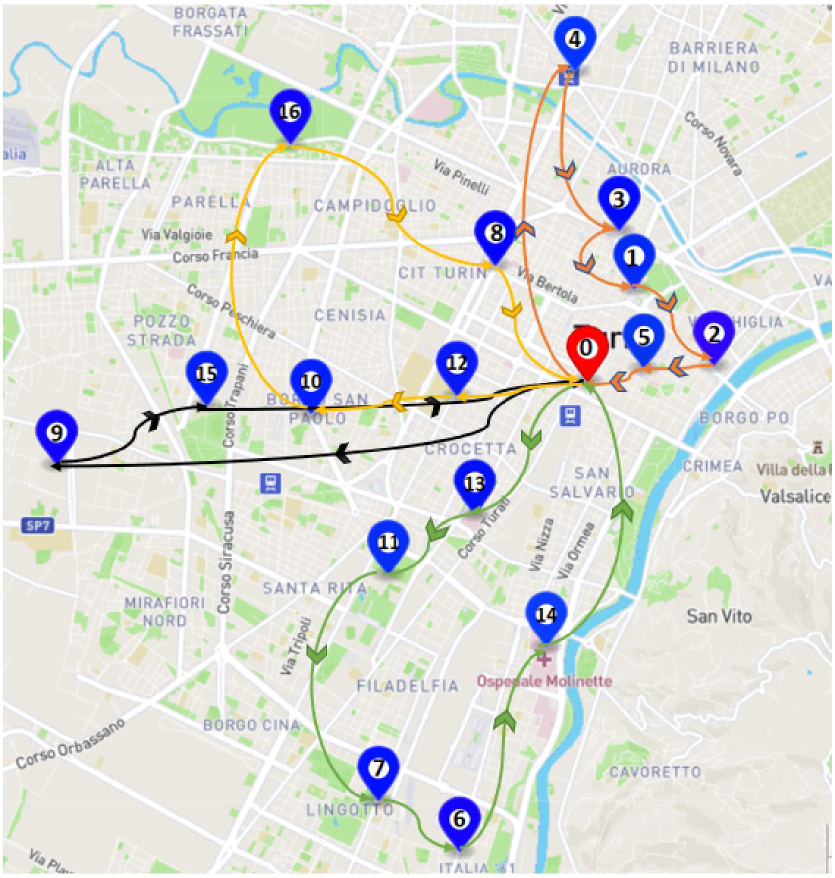


Figure 5.1: Test 5 - routing results no-constraints considered. Parallel Cheapest Insertion applied for first search strategy. The different routes on the map show the connections between points, but not the actual route

Looking at the results shown in Table 5.6, it is clear that the algorithm allocates routes with the least amount of in-flight vehicles conceivable despite having no constraints of any type. The algorithm tries to have all the targets achieved by one vehicle if it is possible concerning the time constraints of each vehicle in flight.

Vehicle 1	
Route	$0 \rightarrow 13 \rightarrow 18 \rightarrow 10 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 17 \rightarrow 3 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 0$
Distance [m]	15373
Time Required	11 min 35 s

Vehicle 2	
Route	$0 \rightarrow 12 \rightarrow 15 \rightarrow 9 \rightarrow 11 \rightarrow 7 \rightarrow 6 \rightarrow 14 \rightarrow 0$
Distance [m]	15383
Time Required	11 min 36 s

Vehicle 3	
Route	$0 \rightarrow 0$
Distance [m]	0
Time Required	0s

Vehicle 4	
Route	$0\rightarrow 0$
Distance [m]	0
Time Required	0s

Vehicle 5	
Route	$0 \rightarrow 0$
Distance [m]	0
Time Required	0 min 0 s

Vehicle 6	
Route	$0 \rightarrow 0$
Distance [m]	0
Time Required	0 min 0 s

Table 5.6: Test 1.2 - routing results no-constraints considered. First solution strategy applied Local Cheapest Insertion

Results	
Maximum of the route distances	30 km and 756 mt
Total time of the routes	0 h 23 min and 11 s
Drone used	2

Table 5.7: Overall results of test 1.2, with first solution strategy applied Local Cheapest Insertion

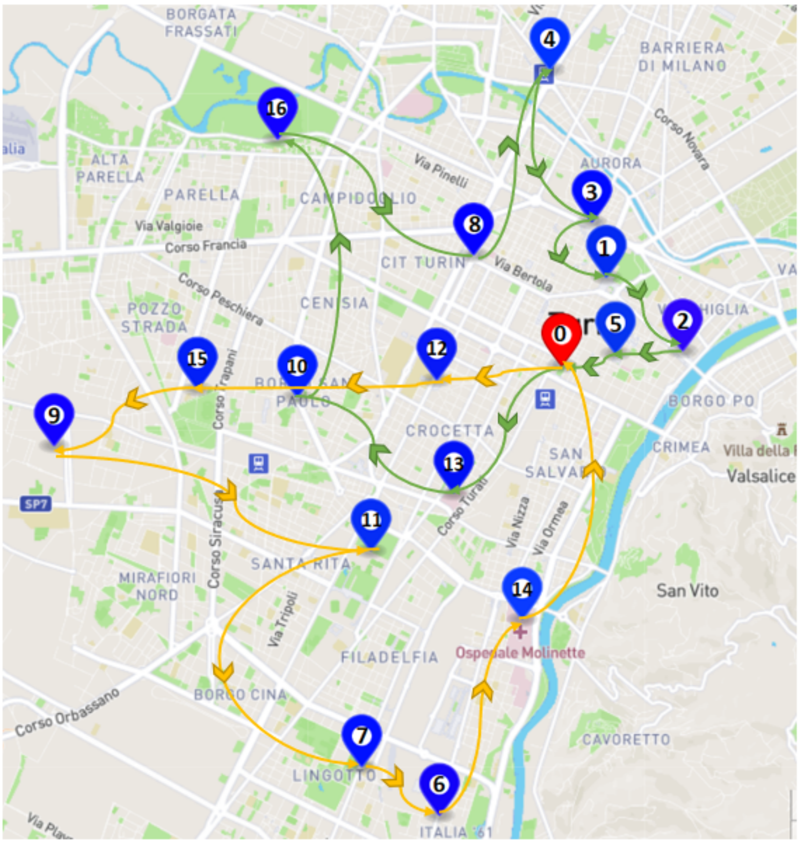


Figure 5.2: Test 1.2 - routing results with no-constraints considered, Local Cheapest Insertion as First Solution Strategy

	Max Route Distance	Max Route Time
Parallel Cheapest Insertion	37 km and 690 mt	27 min and 56 s
Local Cheapest Insertion	30 km and 756 mt	23 min and 11 s

Table 5.8: Comparison between First Local Search strategy applied

Comparing the results, Table 5.8, in an ideal case the application of *Local Cheapest Insertion* as the first strategy solution might be the best. On the other hand, in a real case the waiting time for each passenger would be very high, furthermore. the vehicle's capacity is very restricted and thus such a routing not feasible.

5.2 TEST 2 - VRP with Capacity Constraints

According to Test 2, VRP for air-taxi application study-case has been considered. As for Test 1, the algorithm has been tested on an ideal map of Turin, with the addition of new constraints on **Pickup and Deliveries** and **Capacity**.

In order to be considered a feasible solution for the Air-taxi VRP, it is necessary add also restrictions to Pick up and Drop off passengers at specific locations. The problem thus formulated is associated to **VRPPD**.

In addition, the problem must deal with **Capacity** constraints (**CVRP**): which is the vehicle's maximum passenger capacity, which is set to 2 passengers.

The results obtained (Tab 5.9) have been computed applying a Metaheuristic Algorithm, **Guided Local Search**, imposing as *First Solution Strategy* the **Guided Cheapest Insertion**. The algorithm could obtain a good route planning, but not the optimal one.

Vehicle 1	
Route	DEPOT 0: Load(0) and Time(0s) → 9: Load(1) and Time(316s) → 2: Load(0) and Time(692s) → DEPOT 0 Load(0) and Time (832s)
Distance [m]	13047
Time Required	13 min 52 s

Vehicle 2	
Route	DEPOT 0: Load(0) and Time(0s) → 3: Load(2) and Time(157s) → 12: Load(0) and Time(342s) → DEPOT 0 Load(0) and Time (480s)
Distance [m]	5293m
Time Required	8 min 0 s

Vehicle 3	
Route	DEPOT 0: Load(0) and Time(0s) → 1: Load(1) and Time(130s) → 6: Load(0) and Time(437s) → 0 Load(0) and Time (694s)
Distance [m]	10034
Time Required	11 min 34 s

Vehicle 4	
Route	DEPOT 0: Load(0) and Time(0s) → 5: Load(2) and Time(110s) → 7: Load(0) and Time(407s) → 11: Load(2) and Time(588s) → 18: Load(0) and Time(716s) → DEPOT 0 Load(0) and Time (866s)
Distance [m]	10333
Time Required	14 min 26 s

Vehicle 5	
Route	DEPOT 0: Load(0) and Time(0s) → 8: Load(1) and Time(149s) → 17: Load(0) and Time(320s) → 4: Load(1) and Time(400s) → 15: Load(0) and Time(708s) → 10: Load(2) and Time(833s) → 13: Load(0) and Time(1000s) → DEPOT 0 Load(0) and Time (1150s)
Distance [m]	13031
Time Required	19 min and 10s

Vehicle 6	
Route	DEPOT 0: Load(0) and Time(0s) → 14: Load(2) and Time(188s) → 16: Load(0) and Time(518s) → DEPOT 0 Load(0) and Time (779s)
Distance [m]	11885
Time Required	12 min 59 s

Table 5.9: Test 2 - Routing results with Capacity and Pickup and Deliveries constraints considered

Results	
Maximum of the route distances	63 km and 623 mt
Total time of the routes	1 h 20 min and 1 s
eVTOLs used	6

Table 5.10: Overall results of test 2, with Guided Local Search Algorithm applied

It is important to highlight that during the test, not only the flight time expressed in m/s but also the drone's *descent time* set at 2.5 m/s has been considered. Since the flight height of the drone is set at 100 meters above the ground, descent and ascent time of approximately 80 sec must be added for each route.

At *Depot 0* there is another consideration to be made, in the case of the departure of the drone from depot 0 only the ascent time will be considered while in the case of its arrival, only the descent time of the vehicle, 40 s.

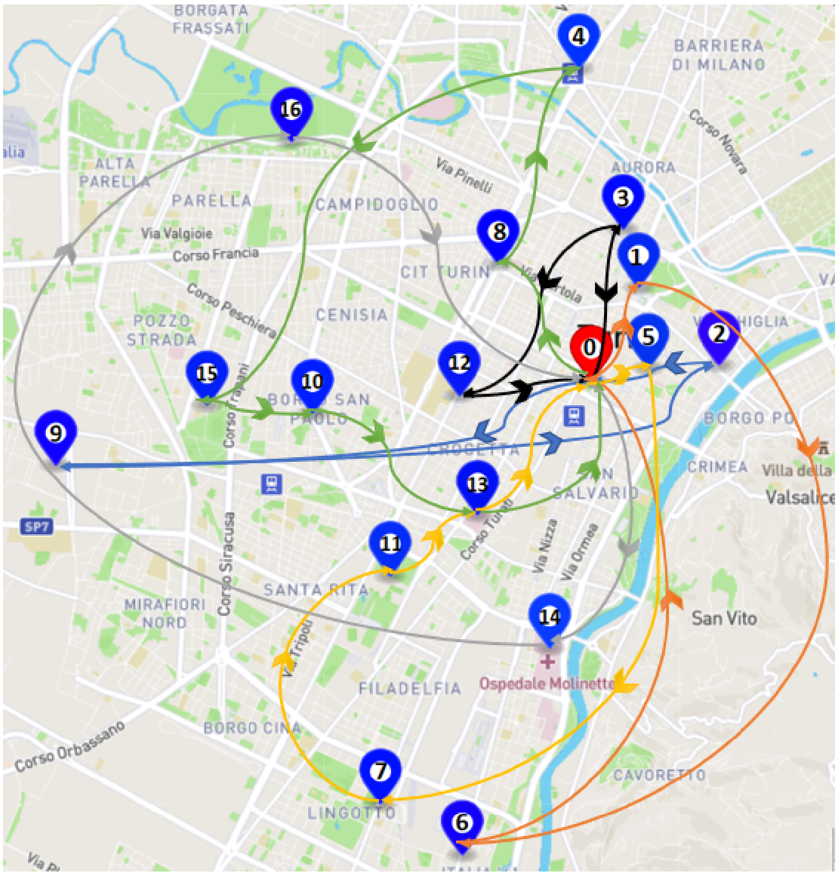


Figure 5.3: Test 2 - routing results with Capacity and Pickup and Deliveries constraints considered

5.3 TEST 3 - VRP with Time Window and Pickup and Deliveries Constraints

In the third test, **Time Windows** constraints have been considered, in order to get closer to a real reconstruction of problem study case. Plus Time Window, the *Capacities and Pickup and Delivers* constraints have been taken into consideration.

To test the effectiveness working of the *Metaheuristics* algorithm, three different simulations have been carried out by setting various time windows constraints.

For the first running test, the Time Windows, chosen for arrival and departure from each Vertistop, have been set as follow:

- **Depot:** [0,0]
- **Vertistops:** [0,600s]

This means that each costumer must be served within 600s, i.e. 10 minutes. It is waiting time for passengers to arrive at the destination Vertistop. Applying such a restrictive choice and running the code, the algorithm finds *no solution* to the problem, as it was possible to expect no routing combination manages to meet the constraints imposed.

On the other hand for the second running test, very broad and nonrestrictive constraints have been applied:

- **Depot:** [0,0]
- **Vertistops:** [0,2000s]

This means that each costumer must be served within 2000s, i.e. 33 minutes and 20 seconds. Looking at the results obtained (Tab 5.11), it can be seen that the algorithm uses as few vehicles as possible to satisfy the routing problem. That is since there are such large constraints on arrival and departure at Vertistops, the routing solution is achievable by applying only 2 vehicles to the air-taxi service. Parallel Cheapest Insertion has been chosen as first strategy applied.

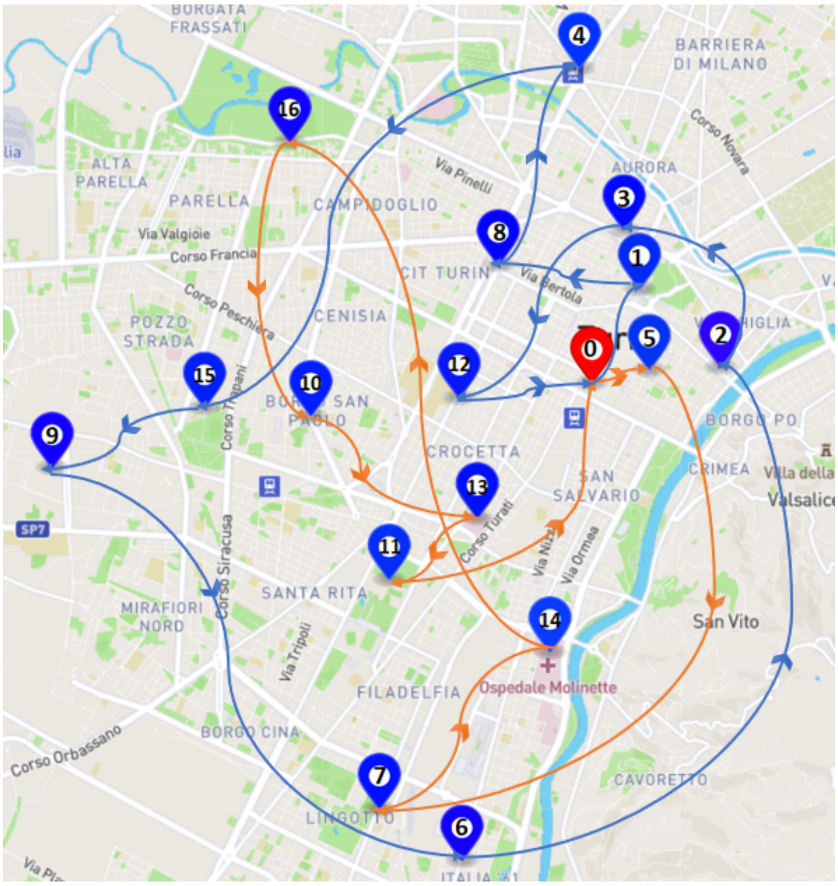


Figure 5.4: Test 3.2 - routing results with Capacity and Time Windows constraints considered, nonrestrictive intervals applied

Vehicle 1	
Route	DEPOT 0: Load(0) and Time(0s) → 5: Time(110s) and Load(2) → 7: Time(407s) and Load(0) → 14: Time(580s) and Load(2) → 16: Time(910s) and Load(0) → 10: Time(1112s) and Load(2) → 13: Time(1279s) and Load(0) → 11: Time(1407s) and Load(2) → 18: Time(1535s) and Load(0) → DEPOT 0: Time(1685s) and Load(0)
Distance [m]	21230
Time Required	28 min 5 s

Vehicle 2	
Route	DEPOT 0: Load(0) and Time(0s) → 1: Time(130s) and Load(1) → 8: Time(271s) and Load(2) → 17: Time(442s) and Load(1) → 4: Time(442s) and Load(2) → 15: Time(750s) and Load(1) → 9: Time(904s) and Load(2) → 6: Time(1209s) and Load(1) → 2: Time(1503s) and Load(0) → 3: Time(1658s) and Load(2) → 12: Time(1843s) and Load(0) → DEPOT 0: Time(1981s) and Load(0)
Distance [m]	24222
Time Required	33 min 1 s

Vehicle 3	
Route	DEPOT 0: Load(0) and Time(0s) → DEPOT 0 Time (0s) and Load(0)
Distance [m]	0
Time Required	0 min 0 s

Vehicle 4	
Route	DEPOT 0: Load(0) and Time(0s) → DEPOT 0 Time (0s) and Load(0)
Distance [m]	0
Time Required	0 min 0 s

Vehicle 5	
Route	DEPOT 0: Load(0) and Time(40s) → DEPOT 0 Load(0) and Time (0s)
Distance [m]	0
Time Required	0 min and 0s

Vehicle 6	
Route	DEPOT 0: Load(0) and Time(0s) → DEPOT 0 Load(0) and Time (0s)
Distance [m]	0
Time Required	0 min 0 s

Table 5.11: Test 3.2 - Routing results with nonrestrictive Time Windows and Capacity Constraints considered

Results	
Total route distances	45 km and 452 mt
Total time of the routes	1h 1 min and 6 s
eVTOLs used	2

Table 5.12: Overall results of test 3.2 with GLS applied

For the last running test, more plausible constraints have applied to the scenario proposed in the problem.

- **Depot:** [0,0]
- **Vertistops:** [0,1000s]

By pointing out that the allowed waiting time for each passenger is set to 120s and vehicles must arrive at the vertiport and take them to destination in a range time of 1000s i.e. 16min and 40s: results obtained as shown in the Table 5.13, are more convincing.

The VRP is not solved by a few vehicles, since the time constraints are more restrictive than those applied in previous test, as it tries to reach the goal in the indicated waiting time, while optimizing the flight time.

Comparing the two tests, it is possible to highlight that the problem that considers large time windows and realizes routing with only two vehicles travels a bigger distance to satisfy all requests with only one vehicle.

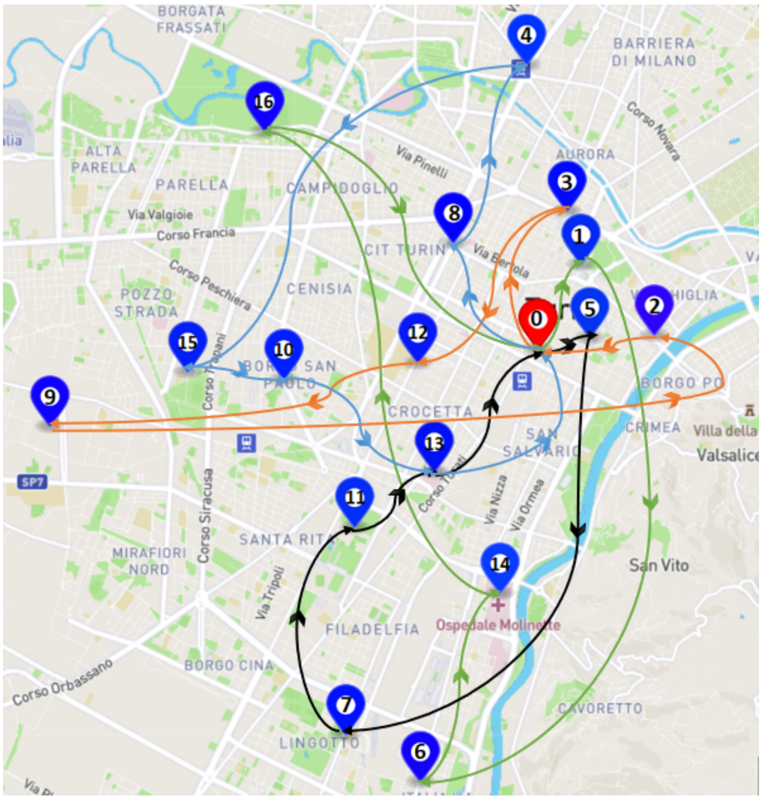


Figure 5.5: Test 3.3 - routing results with plausible Time Windows and Capacity constraints considered

Vehicle 1	
Route	DEPOT 0: Time(0) and Load(0s) → 1: Time(130s) and Load(1) → 6: Time(437s) and Load(0) → 14: Time(587s) and Load(2) → 16: Time(917s) and Load(0) → DEPOT 0: Time(1178s) and Load (0)
Distance [m]	17116
Time Required	19 min and 38 s

Vehicle 2	
Route	DEPOT 0: Time(0s) and Load (0) → 5: Time(110s) and Load(2) → 7: Time(407s) and Load(0) → 11: Time(588s) and Load(2) → 18: Time(716s) and Load(0) → DEPOT 0: Time(866s) and Load (0)
Distance [m]	10252m
Time Required	14 min 26 s

Vehicle 3	
Route	DEPOT 0: Time(0s) and Load (0) → DEPOT Time (0s) and Load (0)
Distance [m]	0
Time Required	0 min 0 s

Vehicle 4	
Route	DEPOT 0: Load(0) and Time(0s) → 3: Time(157s) and Load(2) → 12: Time(342s) and Load(0) → 9: Time(601s) and Load(1) → 2: Time(977s) and Load(0) → DEPOT 0: Time(1117s) and Load(0)
Distance [m]	15774
Time Required	18 min 37 s

Vehicle 5	
Route	DEPOT 0: Time(0s) and Load(0) → 8: Time(149s) and Load(1) → 4: Time(320s) and Load(2) → 17: Time(320s) and Load(1) → 15: Time(628s) and Load(0) → 10: Time(753s) and Load(2) → 13: Time(920s) and Load(0) → DEPOT 0: Time(1070s) and Load(0)
Distance [m]	11220
Time Required	17 min and 50s

Vehicle 6	
Route	DEPOT 0: Time(0s) and Load(0) → DEPOT 0: Time (0s) and Load (0)
Distance [m]	0
Time Required	0 min 0 s

Table 5.13: Test 3.3 - Routing results with plausible Time Windows constraints considered

Results	
Total route distances	54 km and 362 mt
Total time of the routes	1h 10 min and 31 s
eVTOLs used	4

Table 5.14: Overall results of test 3.3, with Guided Local Search Algorithm applied

5.4 TEST 4 - Real case scenario - air-taxi routing in Turin

The last test considered is a real-case scenario for a future air-taxi service within the city of Turin.

Taking into account a risk-map, paths planning have been evaluated associated to Vertistops settled in previous tests, applying RRT* algorithm as explained in Chapter 3.

Data Model obtained are expressed in meters and seconds, as shown in the following **Distance Matrix** (Table 5.16) and **Time Matrix** (Table 5.15).

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
0	0	59	61	81	148	31	298	331	69	247	123	125	63	105	148	170	189	147	100
1	58	0	53	32	110	41	357	371	64	400	157	362	95	127	276	199	226	111	129
2	61	52	0	76	146	31	311	307	116	432	198	335	158	140	162	328	259	146	140
3	83	31	77	0	74	75	398	435	62	372	168	337	106	148	368	209	187	82	149
4	147	111	146	82	0	147	445	488	92	361	203	429	154	327	451	310	151	0	342
5	32	41	30	75	148	0	271	290	87	378	153	326	97	137	146	362	344	147	133
6	290	387	291	385	423	276	0	36	305	265	178	106	173	120	85	289	445	445	120
7	301	410	299	422	466	297	36	0	318	257	191	114	190	135	100	299	459	488	135
8	69	63	117	61	93	88	334	372	0	355	119	157	63	114	199	159	129	92	113
9	239	365	423	376	398	370	256	259	361	0	124	252	189	278	296	84	276	361	282
10	122	157	217	168	208	157	179	183	116	123	0	83	63	99	150	45	122	203	100
11	125	312	354	387	407	288	109	121	156	239	82	0	96	50	73	93	368	429	50
12	66	95	162	107	154	97	174	191	63	182	62	95	0	55	115	106	143	154	55
13	100	129	140	149	342	133	120	135	113	282	100	50	55	0	60	123	343	327	0
14	149	326	164	357	354	149	83	96	205	314	152	75	113	59	0	163	431	451	60
15	170	198	326	210	371	296	291	267	161	83	45	94	106	126	164	0	288	310	123
16	188	222	388	185	141	367	429	404	127	286	123	393	145	381	388	313	0	151	343
17	147	20	146	82	0	147	445	488	92	361	203	429	154	327	451	310	151	0	327
18	100	129	140	149	342	133	120	135	113	282	100	50	55	0	60	123	343	327	0

Table 5.15: Time Matrix obtained through application of RRT* algorithm to a real Turin risk-map

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	1310	1348	1802	3266	698	6569	7282	1537	5439	2717	2756	1395	2314	3277	3744	4160	3252	2213
1	1293	0	1175	7072	2430	913	7866	8170	1412	8806	3462	7964	2098	2808	6088	4382	4976	2454	2855
2	1358	1162	0	1681	3223	689	6859	6764	2569	9506	4369	7385	3477	3085	3570	7228	5717	3223	3084
3	1826	702	1694	0	1643	1652	8762	9574	1367	8198	3698	7415	2343	3275	8110	4608	4127	1821	3286
4	3252	2454	3223	1821	0	3255	9790	10751	2028	7954	4478	9443	3400	7198	9940	6828	3339	0	7542
5	709	915	675	1662	3259	0	5982	6381	1935	8324	3370	7181	2140	3029	3233	7964	7581	3255	2944
6	6399	8515	6423	8490	9316	6078	0	797	6726	5830	3934	2332	3820	2654	1875	6374	9798	9790	2660
7	6632	9041	6578	9301	10270	6549	8002	0	6996	5668	4216	2529	4187	2986	2210	6593	10100	10751	2976
8	1538	1399	2580	1345	2046	1950	7364	8196	0	7810	2626	3462	1397	2511	4387	3516	2848	2028	2505
9	5256	8034	9319	8273	8763	8146	5651	5714	7947	0	2728	5554	4159	6129	6512	1852	6072	7954	6220
10	2687	3471	4778	3710	4589	3463	3939	4045	2552	2723	0	1827	1402	2180	3321	1004	2701	4478	2204
11	2766	6880	7801	8524	8967	6345	2405	2664	3450	5276	1813	0	2119	1108	1614	2060	8112	9443	1119
12	1465	2107	3582	2368	3404	2150	3834	4209	1393	4004	1380	2110	0	1224	2547	2338	3150	3400	1227
13	2213	2855	3084	3286	7542	2944	2660	2976	2505	6220	2204	1119	1227	0	1324	2721	7557	7198	0
14	3287	7181	3615	7868	7807	3279	1835	2119	4511	6927	3355	1661	2500	1314	0	3599	9484	9940	1324
15	3755	4370	7185	4636	8175	6524	6403	5881	3543	1839	1005	2089	2336	2782	3624	0	6338	6828	2721
16	4147	4891	8553	4075	3122	8077	9441	8898	2814	6308	2712	8662	3210	8386	8544	6896	0	3339	7557
17	3252	2454	3223	1821	0	3255	9790	10751	2028	7954	4478	9443	3400	7198	9940	6828	3339	0	7198
18	2213	2855	3084	3286	7542	2944	2660	2976	2505	6220	2204	1119	1227	0	1324	2721	7557	7198	0

Table 5.16: Distance Matrix as a result after haing applied RRT* algorithm to a real risk-map of Turin [m]

The final test of the thesis regards the combination of the constraints on air-taxi operations, that have been evaluated in the previous tests, with the real costs in terms of risk about the city of Turin. Constraints imposed are the following:

- **Capacity Constraints**, set to 2 people
- **Pickup an Deliveries Constraints**, set as in Table 4.2;
- **Time Windows Constraints**, range set to [0,1000s] for each Vertistops as waiting time for passengers to arrive at the destination Vertistop;

Highlighting that, in addition to the time needed to fly from one point to another, the descent and ascent time was added for each Vertisop to 80s and in the depot to 40s.

By setting Parallel Cheapest Insertion as *First Solution Strategy*, the UAVRP problem has been solved by Guided Local Search, metheuristic algorithm, with a running time limit set at 15s.

The results obtained is a good solution for routing problems that regards the combination of the constraints on air-taxi operations, that have been evaluated, with the risk-based distance costs about the city of Turin. The routes have been showed in Table 5.17:

Vehicle 1	
Route	DEPOT 0: Time(0s) and Load(0s) → 3: Time(161s) and Load(2) → 12: Time(347s) and Load(0) → DEPOT 0: Time(493s) and Load(0)
Distance [m]	5566
Time Required	8 min 13 s

Vehicle 2	
Route	DEPOT 0: Time(0s) and Load(0s) → 5: Time(111s) and Load(2) → 7: Time(481s) and Load(0) → 10: Time(752s) and Load(2) → 13: Time(931s) and Load(0) → DEPOT 0: Time(1111s) and Load(0)
Distance [m]	15642
Time Required	18 min 31s

Vehicle 3	
Route	DEPOT 0: Load(0) and Time(0s) → 1: Time(139s) and Load(1) → 6: Time(576s) and Load(0) → 11: Time(762s) and Load(2) → 18: Time(892s) and Load(0) → DEPOT 0: Time(1072s) and Load(0)
Distance [m]	14784
Time Required	17 min 52 s

Vehicle 4	
Route	DEPOT 0: Load(0) and Time(0s) → 8: Time(149s) and Load(1) → 17: Time(321s) and Load(0) → 4: Time(321s) and Load(1) → 15: Time(711s) and Load(0) → 0: Time(961s) and Load(0)
Distance [m]	12342
Time Required	16 min 1 s

Vehicle 5	
Route	DEPOT 0: Load(0) and Time(0s) → 14: Time(228s) and Load(2) → 16: Time(739s) and Load(0) → 0: Time(1007s) and Load(0)
Distance [m]	16874
Time Required	16 min and 47s

Vehicle 6	
Route	DEPOT 0: Load(0) and Time(0s) → 9: Time(327s) and Load(1) → 2: Time(830s) and Load(0) → DEPOT 0: Time(971s) and Load(0)
Distance [m]	16082
Time Required	16 min 11 s

Table 5.17: Routing results of UAVRP tested on a real scenario

Results	
Total route distances	81 km and 290 mt
Total time of the routes	1h 33 min and 35 s
eVTOLs used	6

Table 5.18: Test 3.2 results, with Guided Local Search Algorithm applied

All constraints have been respected, as maximum time for vehicle in flight at constant range speed and time windows, where time constraint of 1000s affects the routing that satisfies the air-taxi service, without affecting the vehicle's route which at the end of service returns to the Depot.

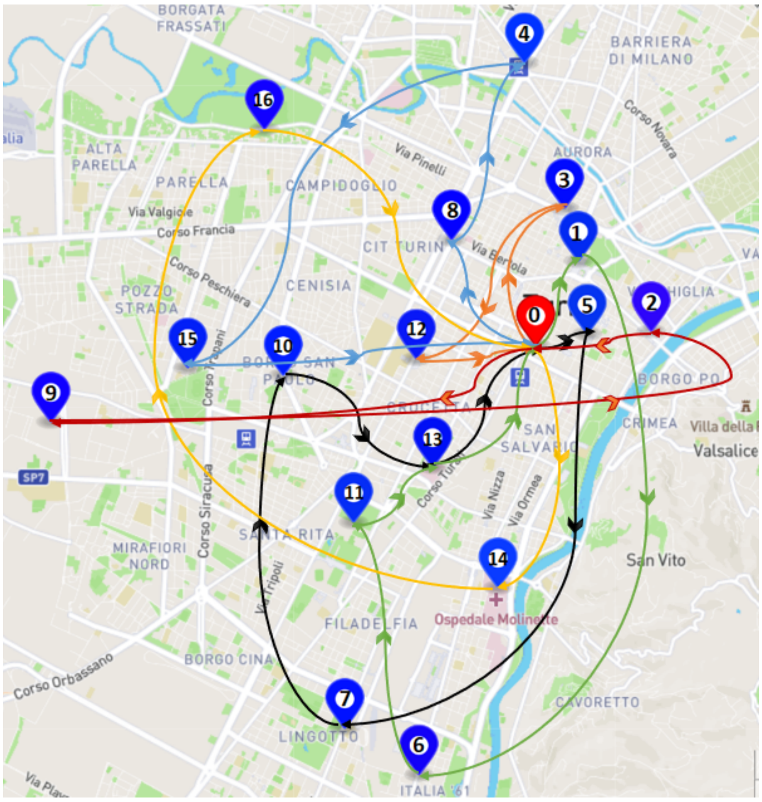


Figure 5.6: TEST 4- Routes results of UAVRP applied to a real scenario

The average time traveled by each vehicle is around 16 minutes. All vehicles have been used during routing in order to reach the goal of minimizing both risk and flight time.

Making a comparison between the results obtained in Test 3.3 where the same constraints have been applied compared to the real scenario test, what varies is the cost that characterizes each route: the fleet of vehicles travels a longer route by going on different paths, but always trying to optimize flight time, trying to avoid the high risk areas of the map.

The sum of the time required for each vehicle to meet the demand for transportation from one point to another of all passengers is equal to 1h 33min and 35s. The set of fleets within the city travels for 123 km and 530mt with 6 vehicles.

In Figure 5.7 are depicted through the Gnatt Chart results obtained. These are considered as optimal values for a UAVRP to which increasingly restrictive constraints have been applied as we went along in order to arrive at simulating a scenario as real as possible. The application of the RRT* algorithm for finding optimal risk-avoiding paths and the application of a GLS meta-heuristic algorithm resulted in routes that simultaneously minimize flight risk and time.

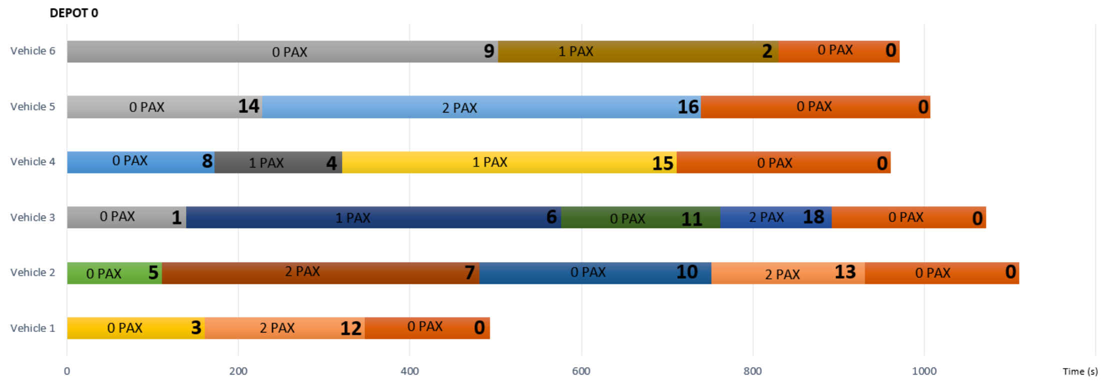


Figure 5.7: Test 4 - Gnatt Chart of UAVRP solution

Chapter 6

Conclusion

According to the need of planning routes for an air-taxis fleet to be safe, short, low-energy, and optimally organized and distributed, several scenarios considering different constraints have been evaluated in this thesis.

Firstly, a simple and ideal scenario minimizing the vehicles' routes in time and distance, based on Turin use-case has been considered evaluating straight routes that connect the Vertiports. In a second instance, the complexity of the scenario is increased by introducing other constraints related to air-taxi operation. The evaluation of these constraints influences the resulting solutions.

In relation to the previous chapter, all the condition have been covered carrying out 4 different tests. Once fixed the previously cited constraints, Guided Local Search a meta-heuristics algorithm explores the objective function related to the routing problem looking for a minimum risk-based distance cost, considered as the optimal path for the selected scenario.

The previously cited tests have been identified as listed below:

- **Test 1**: No constraints considered;
- **Test 2**: VRP with Capacity Constraints;
- **Test 3**: VRP with Capacity, Time Windows and Pickup and Deliveries Constraints;
- **Test 4**: UAVRP with a more realistic scenario, considering all the constraints assumed in previous tests, and using a risk-aware path planning to compute safe routes;

In fact, as a last test, a more complex scenario has been evaluated, by considering Turin risk-map used to compute the minimum risk path using risk-aware path planning algorithm based on the well-known RRT*.

Applying a meta-heuristic algorithm, an optimal solution has been found by minimizing the vehicles' routes in time and distance, as well as considering the operational ground risk to third parties.

The results obtained from the previously cited four tests can be compared in terms of:

- Number of vehicles used;
- Total flight time needed to solve the routing problem;
- Total distance covered by fleet of vehicles.

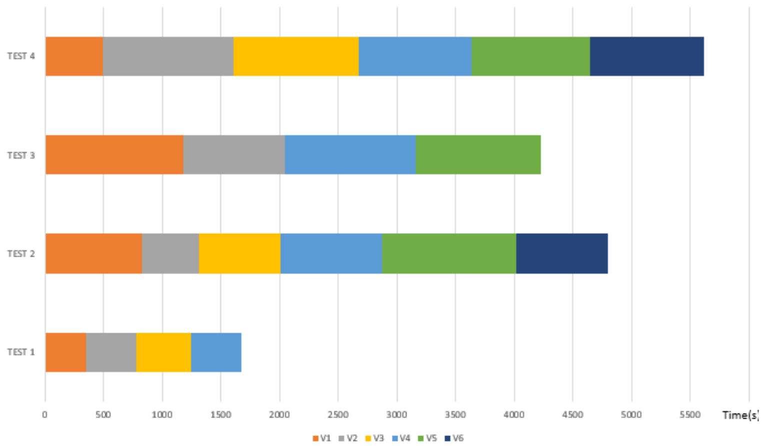


Figure 6.1: Comparison between the results obtained of the total time required for each fleet to perform air-taxi urban service by the four tests. V1 to V6 stand for the vehicles involved in each scenario

As shown in Figure 6.1, all tests occur in different results as the constraints associated to each scenario are different. Considering the increasing level of restrictiveness from scenario 1 to scenario 4, it is reasonable that time, distance and resources involved are increasing.

According to the final test, it can be emphasized that the simulation tools is working and results obtained from the proposed method are satisfactory to deal with an urban air-taxi scenario. In a reasonable time, an optimal solution for the fleet route combination is obtained while combining the overall specifications of both the vehicle and the constraints of time, capacity, and considering the risk to third parties.

As far as the research is developing in this direction, future works can involve additional studies making the VRP scenario the closest to reality as possible.

These studies could be done through a more in-depth study on the possibility of defining a different, and non-constant, speed profile for each route. In addition, it could be interesting to consider how to manage to avoid possible collisions between the various vehicles that could interfere to the urban air-taxi scenario and negatively affect the risk.

Bibliography

[1] Suchithra Rajendran, Sharan Srinivas, and Trenton Grimshaw. «Predicting demand for air taxi urban aviation services using machine learning algorithms». In: *Journal of Air Transport Management* 92 (2021), p. 102043 (cit. on pp. 1, 7).

[2] Anna Straubinger, Raoul Rothfeld, Michael Shamiyeh, Kai-Daniel Büchter, Jochen Kaiser, and Kay Olaf Plötner. «An overview of current research and developments in urban air mobility–Setting the scene for UAM introduction». In: *Journal of Air Transport Management* 87 (2020), p. 101852 (cit. on pp. 4, 9).

[3] Michael D Patterson et al. «An initial concept for intermediate-state, passenger-carrying urban air mobility operations». In: *AIAA Scitech 2021 Forum*. 2021, p. 1626 (cit. on pp. 5, 13).

[4] Parker D Vascik, R John Hansman, and Nicholas S Dunn. «Analysis of urban air mobility operational constraints». In: *Journal of Air Transportation* 26.4 (2018), pp. 133–146 (cit. on p. 6).

[5] *Volocopter VoloPort: The Efficient Ready-Made Vertiport Network Solution for Urban eVTOL Operations - Volocopter — volocopter.com*. `https://www.volocopter.com/newsroom/voloport-efficient-vertiport/`. [Accessed 04-Sep-2022] (cit. on p. 8).

[6] Parker D Vascik and R John Hansman. «Development of vertiport capacity envelopes and analysis of their sensitivity to topological and operational factors». In: *AIAA Scitech 2019 Forum*. 2019, p. 0526 (cit. on p. 8).

[7] Michael Shamiyeh, Raoul Rothfeld, and Mirko Hornung. «A performance benchmark of recent personal air vehicle concepts for urban air mobility». In: *Proceedings of the 31st Congress of the International Council of the Aeronautical Sciences, Belo Horizonte, Brazil*. Vol. 14. 2018 (cit. on p. 10).

[8] Aleksandar Bauranov and Jasenka Rakas. «Designing airspace for urban air mobility: A review of concepts and approaches». In: *Progress in Aerospace Sciences* 125 (2021), p. 100726 (cit. on pp. 12, 15).

[9] Emmanuel Sunil, JM Hoekstra, Joost Ellerbroek, Frank Bussink, Dennis Nieuwenhuisen, Andrija Vidosavljevic, and Stefan Kern. «Metropolis: Relating airspace structure and capacity for extreme traffic densities». In: *Proceedings of the 11th USA/Europe Air Traffic Management Research and Development Seminar (ATM2015), Lisbon (Portugal), 23-26 June, 2015*. FAA/Eurocontrol. 2015 (cit. on p. 14).

[10] Joonas Lieb and Andreas Volkert. «Unmanned Aircraft Systems Traffic Management: A comparsion on the FAA UTM and the European CORUS ConOps based on U-space». In: *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*. IEEE. 2020, pp. 1–6 (cit. on p. 15).

[11] Daniela Rojas Viloria, Elyn L Solano-Charris, Andrés Muñoz-Villamizar, and Jairo R Montoya-Torres. «Unmanned aerial vehicles/drones in vehicle routing problems: a literature review». In: *International Transactions in Operational Research* 28.4 (2021), pp. 1626–1657 (cit. on p. 17).

[12] Mariam Adbelhafiz, Ahmed Mostafa, and Anouck Girard. «Vehicle routing problem instances: Application to multi-uav mission planning». In: *AIAA Guidance, Navigation, and Control Conference*. 2010, p. 8435 (cit. on pp. 18, 19).

[13] Wulan Herdianti, Alexander AS Gunawan, and Siti Komsiyah. «Distribution cost optimization using pigeon inspired optimization method with reverse learning mechanism». In: *Procedia Computer Science* 179 (2021), pp. 920–929 (cit. on p. 18).

[14] Wikipedia contributors. *Volocopter 2X — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-August-2022]. 2022. URL: `https://en.wikipedia.org/w/index.php?title=Volocopter_2X&oldid=1068162389` (cit. on p. 21).

[15] Sin C Ho and Dag Haugland. «A tabu search heuristic for the vehicle routing problem with time windows and split deliveries». In: *Computers & Operations Research* 31.12 (2004), pp. 1947–1964 (cit. on p. 22).

[16] Stefano Primatesta, Luca Spanò Cuomo, Giorgio Guglieri, and Alessandro Rizzo. «An innovative algorithm to estimate risk optimum path for unmanned aerial vehicles in urban environments». In: *Transportation research procedia* 35 (2018), pp. 44–53 (cit. on pp. 25, 26).

[17] Stefano Primatesta, Alessandro Rizzo, and Anders la Cour-Harbo. «Ground risk map for unmanned aircraft in urban environments». In: *Journal of Intelligent & Robotic Systems* 97.3 (2020), pp. 489–509 (cit. on pp. 27, 28).

[18] Mélanie Despeisse and Simon Ford. «The role of additive manufacturing in improving resource efficiency and sustainability». In: *IFIP International Conference on Advances in Production Management Systems*. Springer. 2015, pp. 129–136 (cit. on p. 28).

[19] Sertac Karaman and Emilio Frazzoli. «Sampling-based algorithms for optimal motion planning». In: *The international journal of robotics research* 30.7 (2011), pp. 846–894 (cit. on p. 28).

[20] IA Sucan, Mark Moll, and LE Kavraki. «The open motion planning library (OMPL)». In: *IEEE Robotics & Automation Magazine* (2010) (cit. on p. 29).

[21] Aravind Mohan, Anandhu Dileep, Sreesankar Ajayan, Georg Gutjahr, and Prema Nedungadi. «Comparison of metaheuristics for a vehicle routing problem in a farming community». In: *Symposium on Machine Learning and Metaheuristics Algorithms, and Applications*. Springer. 2019, pp. 49–63 (cit. on p. 33).

[22] Christos Voudouris and Edward Tsang. «Guided local search and its application to the traveling salesman problem». In: *European journal of operational research* 113.2 (1999), pp. 469–499 (cit. on pp. 34, 37).

[23] Wikipedia contributors. *OR-Tools — Wikipedia, The Free Encyclopedia*. [Online; accessed 14-September-2022]. 2022. URL: `https://en.wikipedia.org/w/index.php?title=OR-Tools&oldid=1085017466` (cit. on p. 38).

[24] Wikipedia contributors. *Guided Local Search — Wikipedia, The Free Encyclopedia*. [Online; accessed 1-October-2022]. 2021. URL: `https://en.wikipedia.org/w/index.php?title=Guided_Local_Search&oldid=1054288037` (cit. on p. 40).