UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

PH.D. THESIS
IN
INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

OPTIMIZATION MODELS AND ALGORITHMS FOR UNMANNED AERIAL VEHICLE APPLICATIONS

ADRIANO MASONE

TUTORS: PROF. ANTONIO SFORZA
PROF. LUIGI PAURA

COORDINATOR: PROF. DANIELE RICCIO

XXXII CICLO

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE DELL'INFORMAZIONE

Contents

Introduction

Unmanned aerial vehicles (UAVs) or drones are currently in the prominence of technology expansion of many organizations in different application fields including logistics, surveillance, security, emergency response, agriculture, and investigative science. Indeed, recent advances in unmanned vehicle technology set up new opportunities to increase the efficiency and effectiveness of drone systems that can be exploited by these organizations. However, there are many complex issues in modeling, solving, and implementing drone systems. These issues constitute a topic receiving considerable attention in the last years but the operations research community has only recently started to focus on them with particular reference to the logistic applications.

Indeed, the use of drones for last-mile delivery induced a transformation in the logistic processes since it can produce several benefits to this field: cost reduction if compared to a regular delivery vehicle since a drone operates without a costly human pilot; time reduction since a drone is usually faster than a truck and its route is not affected by the traffic since it can fly over congested roads without delay. However, there are some limitations to the drone use. The size of the drone represents a constraint on the maximum size and weight of the parcels it can carry. Furthermore, the current drone technology can put an upper limit to the number of packages that can be carried simultaneously. Therefore, if the number of customers to be served is greater than this limit, then the drone has to return to the depot to pick up further parcels. Finally, the delivery range is constrained by the drone battery and it can result limited, especially if compared to regular truck delivery.

On the other hand, we recall that a regular delivery truck is heavier and slower than a delivery drone but it also has a long delivery range and can carry several parcels simultaneously (the truck capacity can be assumed to be infinite compared to the drone one).

On this basis, a new combined delivery system has been proposed in literature that consists of the use of a truck and a drone to exploit the advantages of both the vehicles. In this system, the delivery truck and the drone jointly serve all customers. From a transportation planning perspective, this innovative system gives rise to planning problems that can be described as truck-and-drone coordination problems. These problems involve both assignment decisions and routing decisions. Assignment decisions determine which vehicle, drone or truck, will serve which customers, and routing decisions determine the delivery sequence according to which customers assigned to each vehicle are visited.

This thesis brings together an overview of drone technology and applications, and a state-of-the-art of the operational research contributions to the optimization of drone systems. Moreover, it presents some optimization approaches to some truck-and-drone coordination problems. In particular, the outline of the thesis is described in the following.

Chapter 1 presents an overview of drone history and terminology. Moreover, it describes some characteristics of the drone technology and its applications. Finally, a classification of the operational problems arising from these applications is presented together with some drone characteristics that are relevant in the modelling of the related operational problems.

Chapter 2 provides an extensive review of the literature contributions to the optimization problems with drones. The review is divided in two parts. The first part covers the problems arising in application fields as safety, surveillance and covering. The second part collects the problems arising in the logistic industry. Moreover, an original classification of these problems is reported at the end of this chapter.

Chapter 3 presents a general framework for solving truck-and drone-coordination problems based on the concept of operation. An operation is a set of actions performed by both vehicles. The vehicles must be together at the beginning and at the end of an operation. The framework exploits the possibility to represent a solution of a generic truck-and-drone problem in terms of operations and transforms the original problem into the well-known travelling salesman problem.

Chapter 4 introduces a new mathematical formulation for a problem proposed in literature consisting in a truck-and-drone delivery system where the two vehicles work in tandem. Unlike other formulations proposed in literature, the proposed one does not use $BigM$ constraints. Moreover, a Branch-and-Cut algorithm, able to solve to optimality instances of up to 20 customers, is described.

Chapter 5 focuses on a truck-and-drone coordination problem where the truck has the capability of launching and retrieving a drone along the road network edges. A computational analysis is reported which quantifies potential gains achievable by using this capability. The solution of this problem is obtained through an original heuristic solution algorithm which uses discretized and continuous methods.

The studies reported in these chapters were carried out with the members of the Optimization and Problem Solving Laboratory (Professors Maurizio Boccia, Antonio Sforza and Claudio Sterle) of the University Federico II of Naples, Professor Bruce Golden of the University of Maryland and Professor Stefan Poikonen of the University of Colorado. In particular, chapters 3 and 5 are the results of four study and research abroad periods (two periods at the University of Maryland and two at the University of Colorado).
Finally, we highlight that the results of the studies presented in the last three chapters are competitive with or represent the state of the art on the addressed topics. As evidence of this, these studies produced papers that are submitted or will be submitted to some of the most recognized journals of the Operations Research community. In particular, the studies reported in chapters 3 and 5 produced two papers submitted to *Computers & Operations Research* and *INFORMS Journals on Computing*, respectively. On the other hand, the study reported in chapter 4 produced a working paper that will be submitted to *Transportation Research Part C: Emerging Technologies*.

Chapter 1

Unmanned aerial vehicles: history, technology and applications

1.1 Introduction

Use of drones for civil applications gives a new perspective in the socioeconomic system management. In the past, drones were principally employed in the military field for warfare in remote zones. Drone use in the military domain set the groundwork for the use of drones in civil contexts and constituted the basis of the civil market for drones. As a consequence, nowadays small drones for civil use are increasingly available for purchase by private consumers, because they become cheaper and easier to buy. For few tens of euros, small drones can be purchased in electronic toy stores or via the Internet. Moreover, it is possible to buy a professional drone with an advanced camera for taking photo and video with few hundreds of euros. The majority of the non-military drones can fly within a range of a couple of meters to a several hundred meters and have a weight up to several kilograms.

It is virtually possible to use drones in all socioeconomic fields. In the public sector, possible applications can be: crime prevention and crime scene reconstruction, disaster mitigation, dike inspection, geological analysis, fraud detection, border control and environmental and agricultural inspections. In the private sector the potential applications are: aerial photography, heat detection, water transport, package delivery, medicine delivery.

The potential advantages connected to the use of drones are balanced by some threats that can arise from their use: the possibility of drones to be target of damage (e.g., for stealing the drone itself or its payload); the possibility to be an environmental responsible for damaging effects (e.g., air traffic crashing).

In this chapter, in Section 1.2 we will depict a brief history of drones giving a short description of the terminology. Then, some key elements of the drone technology will be depicted in Section 1.3. An overview of the most common civil applications is reported in Section 1.4. A classification of the operational problems arising from drone applications is reported in Section 1.5. Finally, some drone characteristics that are relevant to operational planning are discussed in Section 1.6.

1.2 Drone history and terminology

The term drone was firstly used in military applications and it continues to keep this connotation still today for many people. The earliest unmanned aircraft was probably the steam-powered flying pigeon of Architas the Tarantine in ancient Greece [23] (Figure 1.1). However, this pigeon cannot be strictly considered as the first drone mainly because it was not possible to control its flight.

During the first world war, radio control techniques were employed in the construction of the unmanned aircraft. The Hewitt-Sperry Automatic Airplane flied for the first time in 1917. It was developed as an aerial torpedo and it is considered as the precursor of cruise missile since it was mainly a flying bomb. A proper unmanned aerial vehicle, called Kettering Bug, and able to strike targets within a range of 120 km and flying at 80 km/h, flew for the first time in 1918.

After the first world war, airplanes were transformed into drones. The first examples of this new kind of aircraft were the Larynx (1927), the Fairy Queen (1931) and the DH82.B Queen Bee (1935). In particular, the name Queen Bee is said to have led the use of term ‘drone’ (a male bee) for pilotless aircraft. Indeed, the first use of this term in the US Navy was in 1935, as reported in [18].

The first drone mass-production was during the World War II. Indeed, the Radioplane Company produced nearly 15000 Radioplane OQ-2 drones for the US army. These drones were launched with a catapult and recovered by a parachute.

After World War II, drones were employed for other purposes rather than dropping or being bombs. The MQM-57 Falconer was the first drone used

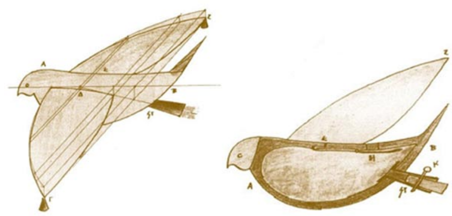


Figure 1.1: A representation of the steam-powered pigeon of Archytas

Figure 1.2: An image of the MQ-1 Predator

for aerial reconnaissance in 1955. Its weight was 124 kg and it could carry cameras and illumination flares for night reconnaissance. Over 73000 units were produced and they were used in nearly twenty countries.

Nowadays, one of the most popular drone is the MQ-1 Predator (Figure 1.2). In 2013 the total number of Predators built was 360 and most of these are still in service. A single Predator drone costs about 4 million US dollars. Predators are longer than eight meters, with a wingspan of about 15 meters and a weight of 512 kg (empty). They can fly at a speed of 130-165 km/h, within a range of 1110 km and an autonomy of 24 hours. It is equipped with cameras and other sensors and can carry fire missiles. It has been used since 1995 by the US Air Force and the CIA for military reconnaissance and combat but later it has been used also for border enforcement, scientific studies and forest fire monitoring.

The interested reader is referred to [21] for a detailed overview of drone history but it is clear from this brief overview that drones were mainly developed in a military context even if nowadays they also offer many civil applications as it will be showed in the following. Indeed, today drones can be relatively small, inexpensive and easily available and, accordingly, the images associated with the word drone are slowly shifting from the military ones to small helicopters, generally equipped with a camera, that are controlled by a smartphone.

However, an important issue when talking about drones is related to the terminology. The term drone typically indicates an aircraft that does not carry an on-board pilot and is instead maneuvered by an operator in a ground control system or is capable to fly autonomously. However, in literature and in practice, there are several terms that refer to drones. For example, the most common terms are Unmanned Aerial Vehicle (UAV) and Unmanned Aeriale System (UAS). The term UAV focuses on the flying platform (and its payload), while the term UAS ia a broader term referring to both the flying platform and the ground station that maneuvers the platfom. In practice, both terms (UAV and UAS) are used to indicate the same aircraft as the term drone. Another term often used instead of drone is Remotely Piloted Aircraft Systems (RPAS). This term is generally used to describe unmanned aerial systems that are remotely controlled by a pilot and it refers more to radio control airplanes and helicopters. It differs from the other terms (drone, UAV and UAS) since it assumes that there is a pilot, so excluding fully autonomously flying aircraft. This is not necessarily the cases of drones and UAVs. Indeed, as mentioned before, there are technological developments that allow drones to fly autonomously, for instance, pre-programmed or self-learning. Therefore, all RPASs are UAVs, but not all UAVs are RPASs and so the term RPAS refers to a subset of drones or UAVs.

1.3 Drone technology

Different types of drones can be identified, according to [61], on the basis of some technical properties: structure, degree of autonomy, size and weight, and power source. These characteristics are important to understand the drone operational aspects as the cruising range, the maximum flight duration, and the maximum loading capacity.

Furthermore, it is important to distinguish between the drone itself (i.e., the flying platform) and the equipment attached to it (the payload). In this context, the drone itself can be considered a flying platform which can be made suitable for different tasks. These tasks can be performed only in combination with a specific payload suitable for that goal. For instance, a camera can be attached to a drone to make it suitable for particular inspections. Different types of

Figure 1.3: An image of the Raven drone

payloads can be distinguished, including freight (e.g., mail parcels, medicines, fire extinguishing material, flyers, etc.) and different types of sensors (e.g., cameras, sniffers, meteorological sensors, etc.).

Moreover, to be able to perform a flight, drones need wireless communication with a pilot on the ground or a need for communication with a payload, like a camera or a sensor. To allow this communication to take place frequency spectrum is required. These requirements for frequency spectrum depend on the type of drone, the flight characteristics, and the payload. The main characteristics that can be considered to differentiate drones will be discussed in the following subsections together with some information about the possible payloads that can be carried by a drone.

1.3.1 Structure

The term structure indicates the technology used to keep the drone flying since this characteristic is also the key factor in the definition of drone shape and appearance. On the basis of this technical property, it is possible to differentiate drones in two main types (fixed-wing systems and multirotor systems) representing the majority of existing drones. Examples of other systems are so-called hybrid systems, which are both multirotor and fixed-wing systems, ornithopters, and drones that use turbo fans.

Fixed-Wing Systems are aircraft that use fixed, static wings in combination with forward airspeed to generate lift. Examples of this kind of aircraft are traditional airplanes and different types of gliders like hang gliders or paragliders. Even a simple paper airplane can be defined as a fixed-wing system. An example of a fixed-wing drone is the widely used Raven drone (Figure 1.3).

Multirotor systems are aircraft that use rotary wings to generate lift and are a subset of rotorcraft. A clear example of a rotorcraft is the traditional

Figure 1.4: An image of the Phantom drone made by DJI

Figure 1.5: An image of a hybrid quadcopter

helicopter. Rotorcrafts can have one or multiple rotors. Drones using rotary systems are almost always equipped with multiple small rotors, which are necessary for their stability, hence the name multirotor systems. Generally, these drones use at least four rotors to keep them flying. A popular example of these multirotor drones is the widely used Phantom drone made by the Chinese company DJI (Figure 1.4).

There are pros and cons connected with these two kinds of systems. On one hand, multirotor drones do not need a landing strip, make less noise than their fixed wing counterparts and can hover in the air. On the other hand, fixed-wing drones can fly faster and are more suitable for long distances than their multirotor counterparts.

Some drones cannot be classified within these two categories. Sometimes because the drone simply is neither fixed-wing nor multirotor, sometimes because the drone has characteristics of both types. Hybrid systems are systems that have characteristics of both multirotor and fixed-wing systems. The hybrid quadcopter (Figure 1.5) is an example of such a drone. This drone uses multiple rotors to take-off and land vertically but also has wings so it can fly longer distances.

Drones that are neither fixed-wing nor multirotor systems are very un-

Figure 1.6: An image of an ornithopter drone

Figure 1.7: An image of a T-Hawk drone

common. An example of such a drone is the ornithopter (Figure 1.6). These drones fly by mimicking wing motions of insects or birds. The size of these ornithopters is often similar to the one of the animal they represent. These small drones are mostly still under development and are not widely used in practice.

Another example of drones that are neither fixed-wing nor multirotor are drones using jet engines. The T-Hawk drone is an example of this sort of drone (Figure 1.7). This drone uses a turbo fan, making the drone look more like an unmanned (hydro)jetpack than a fixed-wing or multirotor drone.

1.3.2 Degree of autonomy

Drones always have a certain degree of autonomy since no pilot is on-board. The autonomy can vary from fully controlled by a remote pilot to full autonomous operations. Therefore, an important distinction within the concept of autonomy is the difference between automatic and autonomous systems. An automatic system is a fully preprogrammed system that can perform a preassigned task on its own. Automation also includes aspects like automatic flight stabilization. Autonomous systems, on the other hand, can manage unexpected situations by using a preprogrammed ruleset to help them make choices. Automatic systems are not able to use this 'freedom of choice.'

The United States Department of Defense distinguishes four levels of autonomy for unmanned systems [61]:

1. A human operated systems. These are systems in which a human operator makes all the decisions regarding drone operation. This system does not have any autonomous control over its environment.

2. A human delegated system. This system can perform many functions without the human supervision. It can perform tasks when delegated to do so, without further human input (e.g., engine controls).

3. A human supervised system. This system can perform various tasks when it is given specific permissions and directions by a human. Both the system itself and the supervisor can perform actions based on sensed data. However, the system can only execute actions within the scope of the current task.

4. A fully autonomous system. This system receives commands input by a human and translates these commands in specific tasks without further human interaction. In case of an emergency, a human operator can interfere with these tasks.

1.3.3 Size and weight

Two noteworthy characteristics that highly differentiate drones are size and weight. The size can vary from drones the size of an insect to drones the size of a commercial airplane. The weight can vary from several grams to hundreds of kilograms.

A classification which distinguishes large drones and small drones, but divides the small drones in multiple subcategories is presented in [18]. The author also adds minimum weight indicators to the drone categories. The lower weight limit of large drones is 150 kg for fixed-wing drones and 100 kg for multirotor drones.

Many countries classify drones on the basis of their weight. For example, the Dutch Human Environment and Transport Inspectorate (ILT) makes a distinction between light drones and heavy drones. Light drones are drones lighter than 150 kg and heavy drones are drones of 150 kg or more.

The development of drones is currently focused on making smaller and lighter drones for the general public. Indeed, a shift can be observed from large drones to smaller drones. This shift can require a new definition of the reference categories and the category parameters. Therefore, a classification which makes a distinction between large and small drones but with different criteria than those mentioned above is proposed in [22]. The authors suggest to use the term large drones for fixed wing drones between 20 and 150 kg and multirotor drones between 25 and 100 kg. Small drones are fixed-wing drones up to 20 kg and multirotor drones up to 25 kg. Within the category of small drones, they suggest to use a subcategory of mini drones. Mini drones can vary in weight from several grams up to several kilograms. These mini drones are mainly suitable for indoor applications and recreational applications.

1.3.4 Power source

The last drone characteristic discussed is the power source. There are four main energy sources:

1. Airplane fuel (kerosene). It is mainly used in large fixed-wing drones.

2. Battery cells. They are mainly used in smaller multirotor drones. These drones are short range and require less operating time than drones using kerosene. These drones are often employed for recreational use, making it more practical for the drone to run on a rechargeable battery cell.

3. A fuel cell. It is an electrochemical device that converts chemical energy from fuel directly into electrical energy. This conversion is efficient and environment friendly since it does not require conversions in thermic and mechanical energy, . Fuel cells are currently rarely used in drones. Only fixed-wing drones can be equipped with such a cell because of the cell's relatively high weight. A major advantage of using a fuel cell is the fact that drones can fly longer distances without recharging.

4. Solar cells. Drones using solar cells are rare in the current drone industry. The rising of new technologies, such as thin film phtovoltaic panels, enable the so-called harvesting (i.e., drones may recharge their batteries during flight in the sunlight). Drones using solar cells are mainly fixed-wing drones. However, these cells are also suitable for many multirotor

drones and small ornithopters due to the low efficiency of current solar cells.

1.3.5 Payload

Theoretically, all kinds of payloads can be attached to drones but the weight and size of payloads can cause some restrictions. Most drones are equipped with cameras by its manufacturer. Other specific payloads can be ordered at drone manufacturers, but the payloads can be attached by the drone users themselves.

An important category of payloads are sensors. Most drones are nowadays equipped with cameras and microphones (they often come by standard when buying a drone). Cameras can be regular cameras but also infrared. Infrared cameras may enable night vision and heat sensing. Other sensors include: biological sensors that can trace microorganisms; chemical sensors ('sniffers') that can measure chemical compositions and traces of particular chemical substances (e.g., radioactive particles); meteorological sensors that can measure wind, temperature, humidity, etc.

Apart from sensors, most payloads involves cargo that needs to be delivered, i.e., mail like letters and parcels, medicines, meals, supplies, and fire extinguishers. In some cases, the cargo is not intended for delivery; examples of such payloads are advertisements (e.g., Objects, banners, ticker tapes, and speakers) and WiFi hotspots. However, we recall that there are technological limits to the size and weight of the cargo that small drones currently can carry.

Other useful payloads include, for example, speakers and light signals for crowd control purposes. More controversial is the use of drones equipped with weapons.

1.4 Drone applications

Many industries can potentially benefit from pilotless technology because it can reduce labor cost. Moreover, drones can operate in dangerous environments that would be inaccessible to humans. Furthermore, pilotless technology lowers the weight of the aircraft, and thus its energy consumption, by making the cockpit and environmental systems (i.e., the systems providing air supply, thermal control, and cabin pressurization) unnecessary. Moreover, drones do not require roads and can, thus, access locations that are difficult to reach by roads.

Possible civilian UAV applications include scientific research, search and rescue,emergency response, traffic control tasks, infrastructure support, aerial photography, forest protection and wildfire monitoring, environmental monitoring, energy and electrical facility monitoring, pipeline inspection, and coast guard support, to name but a few of the possible applications.

In the following, the most promising emerging drone applications will be described. The interested reader is referred to [54] for a detailed survey on civil applications of drones.

Search and rescue

UAVs can be of huge advantage in support of public safety, search and rescue operations and disaster management. In case of natural or man-made disasters like floods, Tsunamis, or terrorist attacks, critical infrastructure including water and power utilities, transportation, and telecommunications systems can be partially or fully damaged by the calamity. These kind of events require rapid solutions to provide communications coverage in support of rescue operations. For example, if the public communications networks are disrupted drones can very effective providing prompt disaster warnings and assisting in speeding up rescue and recovery operations. UAVs can be even more useful since they can cover large areas without ever risking the security or safety of the personnel involved in certain dangerous disastrous situations like poisonous gas infiltration, wildfires and avalanches. Generally, UAVs are used in search and rescue missions for taking high resolution images and videos using onboard cameras to survey a given target area. The images and videos can be used to evaluate the magnitude of the damage in the infrastructure caused by the disaster and to find victims or lost persons. Moreover, drones can also carry medical supplies to areas that are inaccessible to conventional vehicle.

Remote sensing

Drones can be used to collect data from other sensors and send the collected data to ground base stations. UAVs equipped with sensors can also be used as aerial sensor network for environmental monitoring and disaster management. Several datasets acquired from UAVs remote sensors have been used to support the research teams, serving a broad range of other applications.

Construction and infrastructure inspection

There is a growing interest in drone uses in large construction projects monitoring and power lines, gas pipelines and communication towers infrastructure inspection. Indeed, drones do not only provide high resolution 3D aerial recordings at low cost, but also offer safety benefits by replacing human operators for dangerous inspections. Generally, the tasks performed by drones for construction and infrastructure inspections are: asset inspections and data acquisition; data processing with 2D and 3D images; detailed reports of the inspected asset; critical land building inspection (e.g., communication tower); extreme condition inspection (e.g., inspection of onshore and offshore assets).

However, drones have also been used to examine terrain at future construction sites, to track progress at existing construction sites, to inventory the assets, and to regularly inspect facilities as part of maintenance.

Precision agriculture

The use of drones in agriculture is a cost-effective and time saving technology which can help for improving crop yields, farms productivity and profitability in farming systems. Generally, the tasks performed by drones in precision agriculture are: spraying targeted fertilizer and pesticide; providing measurements for the irrigation scheduling; plant and field disease detection; weed detection; soil texture mapping; crop residue cover and tillage mapping; field tile drains mapping; crop maturity and crop yield mapping. Furthermore drones can be employed for gathering data from ground sensors (moisture, soil properties, etc.,).

Road traffic monitoring

Drones can be used as a new traffic monitoring technology to collect information about traffic conditions on roads. Compared to the traditional monitoring devices such as loop detectors, surveillance video cameras and microwave sensors, drones are less expensive, and can monitor large continuous road segments or focus on a specific road segment . The most common tasks for the UAVs employed in traffic monitoring are: stopping vehicle for traffic violations (e.g., the UAV can change the traffc light in front of the vehicle it or relay a message to a specific vehicle); recognition of suspicious or abnormal behavior of vehicles moving along the road; monitoring of pedestrian traffic; incident response; monitoring road conditions; emergency vehicle guidance; monitoring of parking lot utilization.

Communication networks

Drones can be used to provide wireless coverage during emergency situations where each drone is used as an aerial wireless base station when the cellular network goes down. They can also be used to supplement the ground base station in order to provide better coverage and higher data rates for users. The typical tasks for drones used as aerial wireless base stations are the following: assisting a wireless network in providing seamless wireless coverage within the serving area; providing connectivity to backbone networks, communication infrastructure, or the Internet being used as gateway nodes; providing wireless connectivity between two or more distant wireless devices without reliable direct communication links being used as relay nodes; collecting delay-tolerant information from a large number of distributed wireless devices.

Delivery of goods
Drones can be used to deliver food, packages and other goods. For example, in healthcare field, ambulance drones can ship medicines, immunizations, and blood samples, into and out of places that are hardly reachable. They can rapidly transport medical instruments in the crucial few minutes after cardiac arrests. Furthermore, postal and logistic companies have been more and more interested to find new business models of delivery due to the rapid demise of snail mail and the massive growth of e-Commerce. Generally, in drone-based delivery system, a UAV is capable of traveling between a pick up location and a delivery location. The UAV is equipped with control processor and GPS module. It receives a transaction packet for the delivery operation that contains the GPS coordinates and the identifier of a package docking device associated with the order. Once the UAV arrives at the delivery location, the control processor checks if the identifier of a package docking device matches the device identifier in the transaction packet, performs the parcel transfer operation, and sends confirmation of completion of the operation to an originator of the order. This thesis focuses on a combined delivery system constituted by a drone and a truck. As it will be showed in the following chapters, this hybrid system received great attention in the literature of the last years since it allows to obtain several benefits in terms of delivery completion time and levels of emissions.

1.5 Drone Management Operations

As seen in the previous section, drones can be used in several and very different application fields. Therefore, the possible related operations management problems are manifold and cross-cutting. Indeed, similar optimization problems may arise from very different application fields. For example, the traveling salesman problem (TSP) can be used to model a problem where a drone that has to visit multiple locations and it can be found in several fields: in the infrastructure industry, where a drone has to inspect points of interest of a building; in agriculture, where a drone collects information on sample points in a field; in delivery applications, where a drone delivers parcel to several customers. In [45], a classification of the operational problems is provided. This classification distinguish different types of drone operations such as:

- Area coverage, where drones should cover a certain area with a sensor of a limited footprint.
- Search operations, where drones have to find a stationary or moving object.
- Routing for a set of locations, where drones have to visit a discrete set of positions.
- Data gathering in a wireless sensor network, where drones have to collect information from a discrete set of locations while considering communication scheduling and memory capacity constraints.
- Allocation of communication links and computing power to mobile devices, where drones are positioned (or routed) to provide communication links to mobile devices of sufficient quality.
- Operational aspects of a self-organizing network of drones, where a fleet of drones has to perform some task in coordination among themselves to achieve a specific goal.

In the following, each type of drone management problem will be discussed in detail.

Area coverage

In coverage problems, one or more drones equipped with sensors of a limited footprint have to monitor (cover) some area P, which can take different shapes. For example, the coverage path planning problem consists in finding paths of drones equipped with sensors of a limited footprint to cover all points of area P at the lowest possible cost. Another problem related to the planning of area coverage consists in maximizing the information collected from the partially covered area, given some budget constraints. A further problem is the determination of drone positions such that their sensors cover all points of P. Theses sort of problems can arise: in disaster management (as post-earthquake assessment), in agriculture (as in observation of vegetation indexes), and in creating digital terrain maps.

Although area coverage problems with drones are closely related to general coverage problems, some peculiarities may arise and are considered in the problem settings. First of all, since drones perform aerial observations, there is a trade-off between taking pictures from a higher altitude with a larger camera footprint, but lower resolution and higher energy consumption, and taking pictures from a lower altitude with a smaller camera footprint, but higher resolution and lower energy consumption. Secondly, cameras and sensors may be attached to a drone at different orientations (eg, side-aimed cameras vs. directly-downward facing cameras), so that different shapes and positions of the camera footprint relative to the drone are possible. Moreover, more sophisticated sensors can change their orientation during flight. Lastly, because drones, especially fixed-wing drones, may traverse long distances quickly, coverage problems involving nonconvex or disconnected areas gain importance.

Search operations

In the search problem with drones, which can be easily observed in wildlife monitoring and search and rescue applications, a search path for one or several drones must be determined to find an object with an unknown location. Obviously, search problems with drones closely resemble search problems with piloted aircraft. Innovations in modeling and methodology in articles on drones are mainly motivated by applications that are now profitable due to the low cost of drone technology when compared to piloted aircraft. Such applications include monitoring of livestock and environmental monitoring (e.g., icebergs).

Routing for a set of locations

In a number of surveillance and delivery applications, drones have to perform a tour over some set of locations which starts and ends at a depot. The resulting planning problems can be modeled as generalized versions of one of the basic routing problems, such as the TSP or the vehicle routing problem (VRP).

Note that, drones face a problem that some vehicles (e.g, long-distance trucks and electric vehicles) do not face, as drones have to periodically refuel or recharge their batteries at depots to overcome their limited travel range. In contrast to trucks, however, drones may select a battery of the most suitable size for the tour, taking into account that energy consumption depends heavily on the weight of the drone. Drones also fly in 3D space without a road network and may have to make a detour to avoid obstacles or dangerous zones. Another aspect that has to be taken into account in route planning is that drones have a minimum turning radius if they travel at a constant speed.

Data gathering in a wireless sensor network

A wireless sensor network (WSN) is a set of spatially distributed wireless sensors that gather information about the environment and transmit it to a base station. Drones can serve as an additional layer between a network of stationary sensors and the base station: spatially distributed sensors gather information about the environment, and drones gather data from stationary sensors and transmit it or carry it back to the base station. In a number of relevant applications without the use of drones, data gathering from stationary sensors would be slow, very expensive or even impossible. Therefore, the use of drones allowed new solutions for the modeling of WSNs. For example, a drone routing can be exploited to gather quickly data from stationary sensors. In contrast to the routing operations of drones, data gathering operations of drones in WSNs have to respect communication, memory, and data recency constraints. For instance, the communication range is limited and reliability of data transmission depends on the communication distance. Direct information transmission from one node (sensor or drone) of the WSN to another node or the base station is called a hop. Because of the limited communication range, nodes may have to perform a multihop transmission. The limited memory capacity of drones and sensors is another constraint. Overall in these problems, because data collection and transmission consume the limited energy of sensors and drones, a widely used objective function is to maximize the WSN life time, that is, the time interval before the first sensor failure due to energy expiration, or maximal energy consumption among sensors to transmit the collected information. Some problems minimize makespan, total travel distance of the drone and the energy required for the drone's operations.

Allocation of communication links and computing power to mobile devices

To establish connectivity with a drone, a mobile device should be located within its communication range. Therefore, the arising planning problems involve decisions about area coverage. However, in contrast to the area coverage problems involve probability theory and nonlinear equations describing communication constraints. For instance, drones have to allocate communication links (and communication time slots) to mobile devices and select such locations that minimize interference and ensure acceptable levels of signal-to-noise ratio. Different ways of drone employment are possible and so different problems can be conceived. First of all, drones may be assigned to stationary locations and serve as intermediaries to connect mobile devices to macrocell base stations. Secondly, instead of staying in a fixed location, a drone may fly, for example, along a cyclic trajectory. Thirdly, drone-to-drone transmissions may take place so that a mobile device connected to one drone may establish links to a mobile device connected to another drone. Generally, the problem consists in the placement of several drones into stationary positions over some area of interest.

Operational aspects of a self-organizing network of drones

In some drone operations, in which communication is an issue and the base station needs to obtain recent collected information with the shortest possible delays, it is possible to conceive a set of drones as a flying wireless ad-hoc network. A flying ad-hoc network (FANET) is a dynamically self-organizing network of drones that may utilize direct drone-to-drone communication. Direct drone-to-drone communication may have several advantages. First, since the speed of direct data transmission between two nodes is usually faster than the flying speed of a drone, multihop transmissions to the base station may increase the recency of the received information from drones that are out of the direct communication range. It may also free up the limited buffer size of the drone for further data collection. Second, communication between drones is important for flying in formation. Third, users can reduce the payload of some drones and economize on cost by equipping only select drones with the hardware enabling direct long-distance communication with the base station or satellites. Fourth, because of the limited communication range, several drones may be required to provide connectivity to mobile devices.

1.6 Peculiarities of drones in operations management problems

Drones have peculiarities that must be considered in modeling, as reported in [45]. In this section, we briefly summarize common parameters and restrictions used in existing modeling approaches for drone operations.

- **Specifics of motion**. Drones are able to move in 3D space. Autopilots of drones are generally able to maintain flight stability, keep the required altitude, and autonomously land and take off. Nevertheless, certain specifics of drone motion may need to be taken into account in planning of drone operations. One of them is the minimum turning radius restriction while changing directions in flight, which is especially important for fixed-wing drones. Although rotorcraft drones, such as quadcopters, may easily change their flight direction by making sharp turns, each reversal requires additional time and energy, as the drones must come to a halt before moving in a different direction. Small and micro drones are highly susceptible to weather conditions, such as wind, which may be modeled as uncertain travel times. Finally, requirements for minimum and maximum flight angles of fixed-wing drones should be taken into account during landings and takeoffs.

- **Limited payload**. The maximum weight of the payloads for package delivery generally does not exceed 3 kg (6.5 pounds), and a drone usually carries just one package per sortie. Limitations on the payload are closely related to the capacity of the drone's energy storage unit and the size and configuration (and cost) of the drone. For example, to maintain a stable flight, the propeller of a rotorcraft drone should generate enough lift to counter the force of gravity. Therefore, a heavier drone needs more energy than a lighter drone to fly the same distance.

- **Limited flight range**. Most drones carry an energy unit of a limited capacity. Energy consumption of a drone depends on a multitude of factors, such as drone structure (fixed wing vs. rotorcraft), flying altitude (e.g., propellers of rotorcrafts have to rotate faster at higher altitudes because of lower air density), flight conditions (such as hovering vs. forward flight), climbing speed, payload, and weather conditions, such as wind. The limited capacity of the energy unit is usually modeled as maximal operation time, maximal flying distance, or the limited number of locations a drone can visit during one flight. Battery swaps and refueling usually require assistance of a human operator; however, there exist fully automated platforms able to exchange or to recharge the battery of a drone in just a couple of minutes.

- **Specifics of information processing and connectivity**. Drones have to maintain communication links with the ground control station to receive instructions and transfer the collected information. Since line-of-sight communications are typically required, the signal gets weaker in the shadow of buildings in urban areas, indoors, or under the crowns of trees. Additionally, transmission lines and telecommunication towers may cause signal interference. Therefore, path planning methodologies may avoid or penalize visitation of certain regions. Drones may use different wireless access methods to provide communication services, which may require assignment of particular time slots and/or frequencies to the users. Another consideration is that the power density of the signal reduces as it passes through a communication channel. Therefore, drone positioning as a flying base station depends on signal fading along the communication path, path loss, interference, and noise. Other connectivity challenges emerge when several drones perform tasks cooperatively, since they may need to exchange information by establishing communication links that are subject to noise and dependent on transmission distance. Drone-to-drone communication also enables drones to attend a GPS-denied area while maintaining a communication link with another drone able to receive the GPS signal. Finally, the limited memory capacity of the drone should be respected in gathering data from sensors.

- **Handling by a human operator**. Drone regulations foresee the compulsory presence of a human operator in several countries. Generally, a human operator performs a number of setup operations before the drone's takeoff and, after its landing, he or she may have to control the drone and to examine information collected by the drone in real time.

1.7 Conclusions

An overview of Unmanned Aerial Vehicles history (starting from when the concept was conceived in ancient Greece), technology (all different kinds of developed UAVs and the correspondent uses) and applications (linked with the different technologies) has been provided in this chapter. This overview represents the starting point of the following chapters. More precisely, the literature contributions on drone applications arising in surveillance and logistics sectors will be analyzed in the next chapter; the studies reported in the successive three chapters are based on the drone management operations and peculiarities discussed in this chapter.

Chapter 2

Literature review

2.1 Introduction

In this chapter, an extensive literature review of operations research contributions related to optimization problems with drones is provided. The review is organized on the basis of the application field of the papers. Section 2.2 is devoted to the studies that tackle optimization problems for surveillance and safety missions (i.e., covering, targeting, image sensing etc.). Section 2.3 is related to works that tackle optimization problems arising in logistics (i.e., last-mile logistics, delivery with drones, etc.). A new classification of this works is provided at the end of the chapter together with an original notation for this kind of problems.

2.2 Surveillance with drones

In this section, a collection of papers tackling problem of surveillance in many application fields (safety, infrastructure inspection, communication networks) is reported. The papers will be presented in chronological order. These papers address different operations management problems as node routing, arc routing, covering and location problems.

UAV Routing for Area Coverage and Remote Sensing (Avellar et al., 2015)
A methodology for optimal time coverage of ground areas using multiple fixed-wing UAVs is presented in [4]. The authors solved the coverage problem by creating a graph and then transforming the original problem into a vehicle routing problem. In particular, they decomposed the area to be covered as a

set of sweeping rows. These rows form the edges of a graph where an original variant of the vehicle routing problem (VRP) is solved. They assumed that the number of human operators responsible for launching and retrieving the UAVs is smaller than the number of vehicles. This assumption is incorporated in the model by defining a so-called setup time. This means that, since one operator cannot prepare more than one UAV at the same time, the setup time of each UAV is cumulative. Given the constraint on the number of operators, there are scenarios where launching a large number of UAVs may have a negative impact in the total mission time, due to the influence of the cumulative setup time. The solution method was evaluated in a real-world experiment with two actual aerial vehicles and a single human operator.

The mobile target covering problem (Di Puglia Pugliese et al., 2016)
A problem where a set of mobile targets (points that have to be monitored such as vehicles, animals, humans) with little a-priori information about their mobility has to be covered through the deployment of UAVs is tackled in [28]. The aim is to ensure that each mobile target is covered by at least one UAV. The authors also add another dimension bcause each UAV can change its observation (coverage) radius, depending on its altitude, to cover more or less targets. In addition, it is assumed that the energy consumed by each UAV is related to its altitude and when an UAV runs out of battery, it is replaced by a new one. The objective is to minimize the number of used UAVs to cover all the targets. The number of UAVs depends on the number of targets, their dispersion, their movement but also on the energy consumed by each UAV. A mixed integer non-linear programming formulation is provided together with a series of valid cuts. The authors also developed a MIP-based heuristic procedure. The proposed solution strategy is an iterative procedure, considering, at each step, a subset of targets to be monitored (i.e., solving a restricted MIP problems), which is enlarged at each iteration. The subproblems are easier to solve than the entire problem and the solution obtained at some iteration is built by considering the decisions taken in the previous ones. The computational results underlines the difficulty of the problem and for only 23 % of the instances the solver has been able to provide the optimal solution in 2 h. The heuristic showes very promising performance, exhibiting a reasonable trade-off between quality of the solution and computational effort.

Drone placement and cost-efficient target coverage (Zorbas et al., 2016)
A drone location problem devoted to determine the optimal location of a set

of drones to cover a set of target is tackled in [65]. The authors assumed drones equipped with one or more electrical motors (quadcopters) and a fixed-angle camera targeting on the ground. The drones are able to identify static or mobile ground targets, which are considered as points that have to be monitored. In case of mobile targets, no a priori information about their mobility is known, except their maximum speed. The aim is the optimal deployment of drones, ensuring, at the same time, that each target is covered by at least one drone. Another dimension in the considered problem is that each drone can change its coverage radius, depending on its altitude that allows it to cover more or less targets. It is assumed that the energy consumed by each drone is related to its altitude. An empirical energy consumption model based on some real measurements with electrical motors and drone manufacturers data is taken into account. The problem objective function is the minimization of the cost, that is the number of drones or the total energy consumption. The maximum number of drones depends on the number of targets, their dispersion, and their movement. An integer and a mixed-integer non-linear optimization models are formulated to tackle the variant with static and mobile targets, respectively. Two heuristic algorithms which provide scalable and efficient solutions to the drone location problem are proposed. The heuristic algorithms can solve instances of the considered problem with more than 50 targets and infinite possible positions for the drones. On the other hand, the models can be solved for up to 10 targets and 7803 possible positions for the drones.

Drone arc routing problems (Campbell et al., 2018)
The idea of using drones to optimize the coverage of specific edges in a network like for inspection sensing and surveillance along linear structure (e.g., roads, railroads, boundaries, pipelines, etc.) is presented in [12]. They define the Drone arc routing problem as follows: given a network and a set of lines that has to be covered, each line with an associated cost, the objective is to determine the drone tour covering all the required lines minimizing the total cost. The main difference with respect to the typical arc routing problems is the drone ability to travel directly between any two points in the plane without considering the edge network. The authors tackle the problem variant which considers the use of a single drone with an unlimited endurance. A heuristic solution method was developed to solve the problem. It iteratively discretizes the network and solves a Rural Postman Problem on the discretized network. The best solution obtained is choosen as the solution of the original problem.

Preliminary results showed that using drones is possible to obtain a saving of 3% on the total cost compared to the one obtained using vehicles that have to follow the network edges.

A vehicle routing problem arising in UAV monitoring (Zhen et al., 2019)
The study reported in [64] investigates a routing problem in which UAVs monitor a set of areas with different accuracy requirements. The main difference with classical vehicle routing problem consists in determining not only the order in which to visit a set of nodes but also the height at which to visit them. In particular, the height will impact directly the accuracy level and the service time. To tackle the problem, the authors discretize the monitoring area into several smaller squares. Each square represents a unit monitoring area. The center of discrete squares or the vertices of the square are the planar projections of the nodes in the space at different heights. On this basis, the problem was formulated through an integer linear programming model minimizing the total duration of the monitoring task. To solve large scale instances, the authors proposed an original Tabu Search metaheuristic method. Computational results showed that the algorithm was able to effectively solve instances up to 81 areas to be monitored by 724 possible points. An illustration of two UAV's monitoring routes is reported in Figure 2.1

Minimizing dispersion in multiple drone routing (Dhein et al., 2019)
The problem addressed by [27] consists in defining the routing of a set of drones deployed to perform a collaborative mission. The spatial and temporal proximity of the drones throughout their route is a crucial factor for the success of the mission (e.g., communication, coordination and situation awareness requirements). The drones must depart simultaneously from the depot, reaching and servicing a predefined set of locations, before returning to the depot. The sequence of locations visited by each drone is chosen minimizing the dispersion with regards to the route of each other drone. A mathematical programming model is proposed to solve the problem but its practical relevance is limited to small-sized instances. Therefore, the authors proposed a Local Search Genetic Algorithm where the genetic algorithm provides a good exploration of the solution space while the local search improve the solutions/elements of the population. In particular, the local search is based on the Variable Neighborhood Search (VNS) framework [40]. It uses two different kind of neighborhoods: one is designed with the objective of reducing the completion time and the other one with the objective

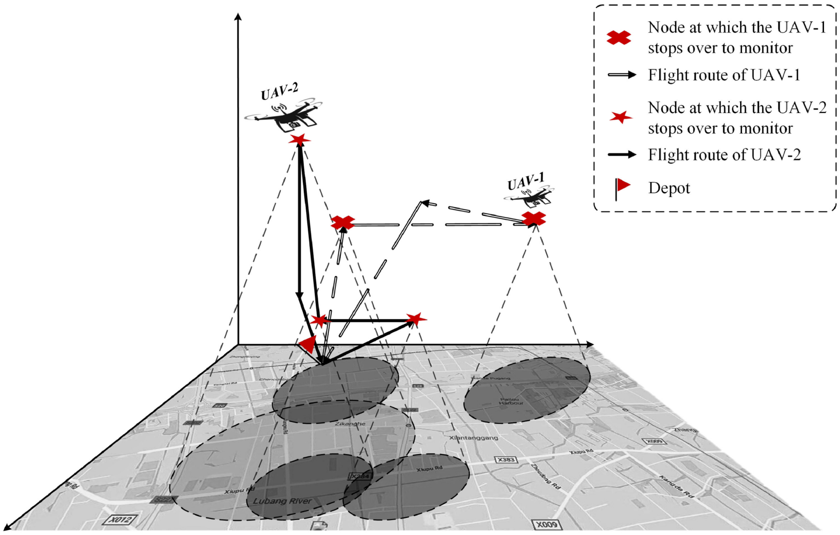


Figure 2.1: An illustration of two UAV's monitoring routes (Source: Zhen et al., 2019)

of improving the synchronization among the drones.

The location-allocation problem of drone base stations (Cicek et al., 2019)
The location problem of multiple Drone Base Stations (DBS) servicing the users in a wireless communication network is tackled in [17]. A sample representation of a telecommunication network using DBSs is reported in Figure 2.2. In this problem, the drones are used as a base stations since they can extend the coverage, improve spectral efficiency and increase the quality of experience. The objective is to determine the number of DBSs and their 3D locations to serve all customers together with the allocation of the available resources among the users to maximize the profit of a service providers. The authors proposed a mathematical model together with a two phase heuristic solution method. In the first phase, it fixes the location of the DBS and solves the bandwidth allocation sub-problem considering the capacity and the service provision. In the second phase, the location of the DBS is modified with respect to the allocation made in the previous phase. The two phases are repeated until the DBS locations remain unchanged in two consecutive iterations.

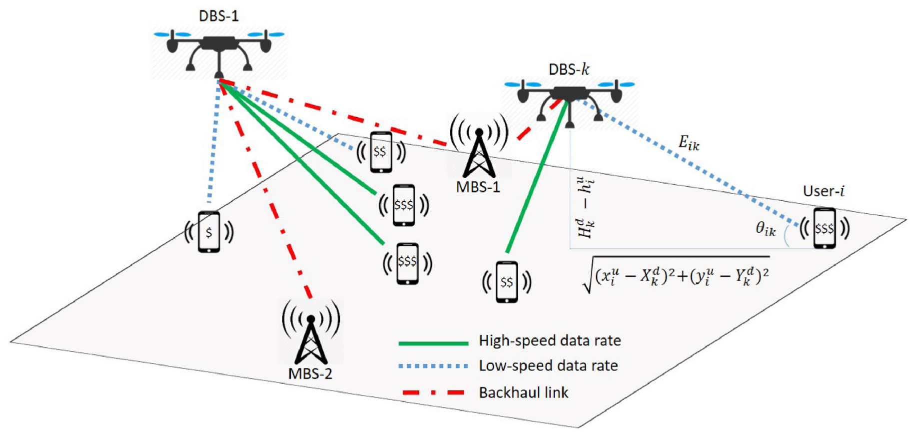


Figure 2.2: An illustration of a telecommunication network using DBSs (Source: Cicek et al., 2019)

Optimization of UAVs coordination in target search (Alfeo et al., 2019)
The problem of discovering of targets located in an unstructured environment, with no prior knowledge about their location and about obstacle layout was tackled in [2], that is . This problem is known in literature as the target search problem whose objective is to minimize the overall time needed for completing the mission. The target search mission is performed by a swarm of UAVs. In the pape,r the swarm behavior is modeled considering two different paradigms: biological behavior like in other social animal metaheuristics and computational behavior which exploits the additional information provided using the UAV techonology. These behaviors are considered in the development of an evolutionary algorithm. The solution of the algorithm is used to determine the parametrization of the stygmergy and of the flocking of the swarm of UAVs. Experimental results showed that considering the UAV techonology in modelling the agent behavior improves the swarm cooperation and coordination.

UAVs for the internal security of a heritage site (Boccia et al., 2020)
The use of drones for the internal security of the the Archeological Park of Pompeii (PAP) is evaluated in [6]. This study arise from a research agreement devoted to identify and apply Operational Research models and methods to support the camera surveillance system already implemented in the park. First, a graph representing the area object of study is obtained through

discretization. Then, a visibility and covering analysis has been carried out based on the approach presented in [28] for different values of drone flying altitude. For each analysis, a coverage matrix is determined representing the input of a Set Covering Problem that is solved using a commercial MIP solver. The drone locations determined in the set covering optimal solutions are used to define a surveillance route for the drone solving a TSP among them. The computational experiments show the effect of the flying altitude and the number of available drones on the time required to monitor the whole park

2.3 Logistics with drones

A selection of studies investigating the use of drone in logistics is reported in this section. The majority of the problems assumes the use of a truck in combination with a drone to overcome some of the drone technical limitations (e.g., limited flight range, limited payload, etc.). In particular, two kinds of coordination problems arise when the two kinds of vehicles are used. The first one foresees that the truck and the drone serve customers independently. The second one requires a certain level of synchronization between the two kinds of vehicles since the drone is carried by the truck when the first one is not servicing some customer.

On this basis, the papers are classified in three categories:

- Vehicle Routing Problems with drones
- Vehicle Routing Problems with trucks and drones in parallel
- Vehicle Routing Problems with trucks and drones in tandem

The contributions belonging to each category will be presented in chronological order in the next subsections. More details about the settings of each problem will be given within each study description.

2.3.1 Vehicle Routing Problems with drones

In this subsection we report the contributions that tackle problems considering the use of drones without the interaction with other kind of vehicles. Generally, these contributions extend the classical settings of vehicle routing problems to take into account drone peculiarities.

A multi-objective green UAV routing problem (Coelho et al., 2017)
A multi-objective Green UAV Routing Problem (GUAVRP) is proposed in [19]. It minimizes seven objective functions: total traveled distance; uavs maximum speed; number of used vehicles; makespans of the last collected and delivered package; average time spent with each package; and maximize batteries load at the end of the schedule. Furthermore, the model respects drones operational requirements, such as: maximum weight they are able to carry, battery minimum Depth-of-Discharge (DoD); UAV maximum speed. Moreover, UAVs are allowed to refuel/charge at the charging stations, since their autonomy does not allow them to fly over long periods. Finally, UAVs can be limited to fly only at predetermined levels of altitude, related to their load capacity and size. A simple flying environment is modelled with the biggest and fastest ones flying at higher levels. These levels or layers can be interconnected by vertical displacement points, where UAV will be allowed to exchange layer and products. Thus, loads can be redistributed by smaller drones using supporting spots that allow vertical displacement between layers (an example of this supporting spot is reported in 2.3). A case of study composed of an airspace divided into two layers is designed: a lower layer in which smaller UAVs travel with lower speed and an upper layer where the traffic is mainly composed of faster drones with heavier loads. Each layer is subdivided into horizontal and vertical strips, where vehicles are allowed to move.

Formulations and Algorithms for Drone Routing Problem (Cheng et al., 2018)
Two formulations to solve the multitrip vehicle routing problem where the vehicles are UAvs are provided in [16]. Both the formulations explicitly consider the influence of payload and distance on flight duration. The difference between this two formulations consists in one having a drone index while the other one doesn't use a drone index. Some valid inequalities based on the energy function are proposed together with a Branch-and-Cut algorithm. Moreover, an extensive literature review on drone routing problems is provided with a table which classifies the surveyed works. A new set of benchmark instances for this problem was generated on the basis of instances used in literature for VRP and VRPD. The extensive numerical experiments showed that the 2-index formulation can solve more instances to optimality and provide good quality solutions.

Figure 2.3: An example of supporting spot that allow drone vertical displacement between layers (Source: Coelho et al., 2017).

A facility location problem for a drone system (Shavarani et al., 2019)
In [55] an optimization problem is presented which considers a drone delivery system comprised of warehouses and refuel stations. The warehouses are the initial take-off locations for the drones. They are considered as the main nest for the drones. Opening a launch center requires more infrastructure in comparison with the refuel stations. The refuel stations are designed for extending the coverage of the drones by giving them the possibility to refuel/recharge on their way to demand points. The capacity of each facility is determined by the total number of assigned drones. A drone may visit one or more refuel stations on its route to satisfy demand. Moreover, the waiting time of a customer to be served is limited by a threshold parameter. Four major costs are addressed in this problem: establishment costs of launch stations, establishment costs of refuel stations, and procurement and usage costs of drones. The authors proposed a fuzzy mathematical formulation to model the problem where the fuzziness is associated to the four considered costs that are minimized in the objective function. Due to non-linearity of the proposed model even small instances were unsolvable. Therefore, the authors proposed a genetic algorithm in which each element of the population represents a feasible solution. A greedy search is applied when generating

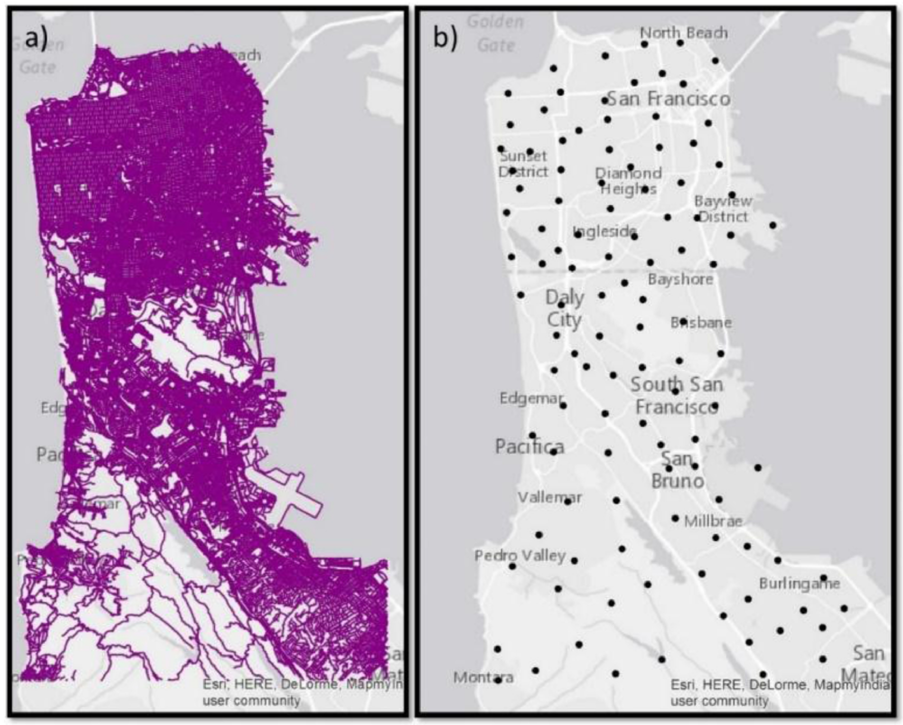


Figure 2.4: (a) - San Francisco's transportation network. (b) - San Francisco's take-off and rendezvous location set (Source: Shavarani et al., 2019).

new elements of the population that optimizes the solution downgrading all the opened facility (from a warehouse to refuel station or from a refuel station to a closed facility) if the objective function value associated to that solution does not increase. The solution method was tested on a case study based on the Amazon Prime Air project applied to the city of San Francisco. The San Francisco's transportation network and candidate location set are shown in Figure 2.4.a and 2.4.b , respectively.

Potential estimation of delivery by drones (Aurambout et al., 2019)
An estimation of the market viability of a drone delivery systems in the EU region is provided in [3]. The considered delivery systems is based on the use

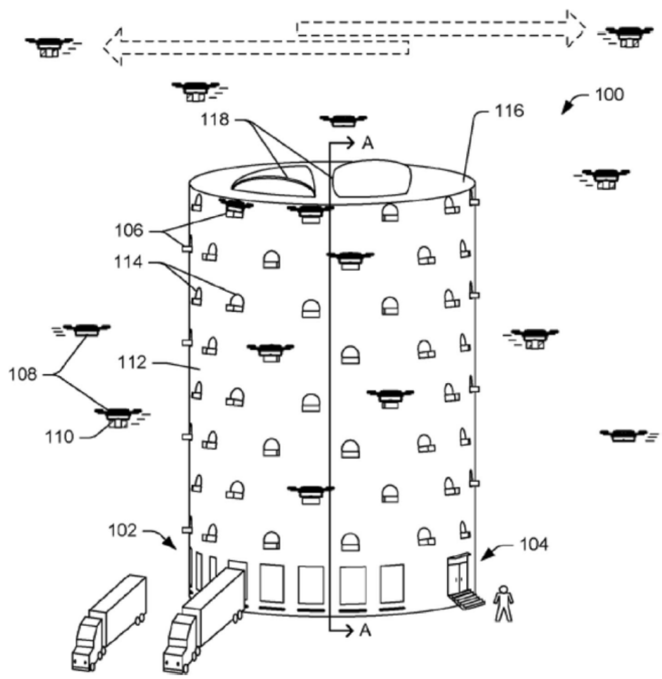


Figure 2.5: Amazon drone-beehive concept (Source: Aurambout et al., 2019)

of fulfillment centres called beehives (Figure 2.5) designed to accommodate landing and take-off of unmanned aerial vehicles in densely populated areas. Drones are used for delivery and require a surface of "open space" to land (gardens, etc.). The drones can travel a maximum distance of 24 km. The set of beehive potential locations and the set of customer locations are defined on the basis of the data produced by the European Commission Joint Research Centre's LUISA (Land Use and Scenario modeling for Integrated Sustainability Assessment) Territorial Modelling Platform. The evaluation of the market potential is performed in a two step modelling approach. The first step computes the potential economic return achievable locating a beehive in a potential location and which deliver each customer within its reachable range, for each location. The second step identifies the highest economic return locations for the beehives on the basis of the whole set of locations. The results of the analysis showed that about the 7.5% of the EU population could benefit from this service.

Pickup and Delivery with autonomous vehicles (Ulmer and Streng, 2019)
The work presented in [58] considers a depot, a set of capacitated pickup stations and a delivery fleet autonomous vehicles. The customer requests are stochastic and follow a known probability distribution. The goods can be delivered to the preferred pickup station or to one in its neighborhood. The provider has to decide about sending a vehicle (or to wait), what to load on a vehicle, and to which station to send the vehicle. The objective is to minimize the delivery time. This problem is defined as the Stochastic Dynamic Dispatiching Problem for Same Day Delivery with Pickup Stations and Autonomous Vehicles. The authors proposed a policy function approximation approach to solve the problem. The parameter considered for the policy is the number of parcel that the vehicle can deliver to a station. If the number of parcel is greater than the selected threshold the vehicle is sent for the delivery. The approach is tested on a real case study for the city of Braunschweig. Computational results showed that the number of deliveries performed using the proposed approach is significantly more than the usual number of deliveries reported in the same-day delivery literature.

A dynamic algorithm for on-demand meal delivery (Liu, 2019)
An investigation on the online routing of drones for meal pickup and delivery in a dynamic operational environment is reported in [38]. In this context, it is assumed that meal orders are requested randomly over the time horizon and the corresponding delivery location is random across the considered region. The objective of the fleet operator is to dispatch the drones in real-time to make the fastest and most efficient delivery of all orders. The author presented a an optimization-driven progressive algorithm for drone dispatch and order delivery in a dynamic, real-time operational environment. The algorithm is based on a mathematical model of the business operations and a temporal progression framework that connects decision across time periods. Computational and simulation experiments showed the algorithm capability of handling online dispatches of moderately sized systems.

2.3.2 Vehicle Routing Problems with trucks and drones in parallel

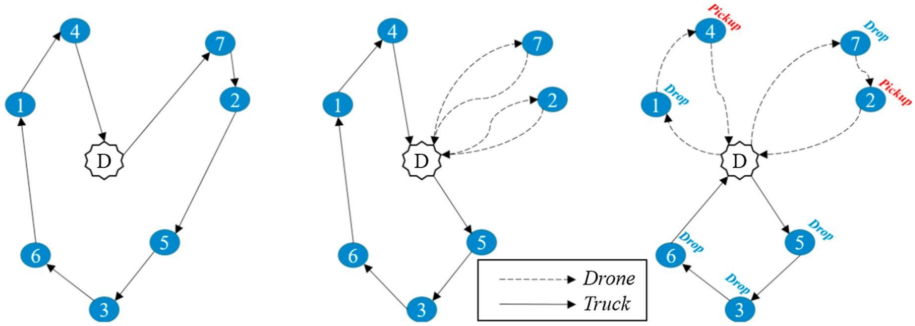
In this subsection we report the contributions in which trucks and drones operate in parallel. In these problems, the required level of synchronization between the two vehicles is low. The main decision consists in the assignment of the customers to each kind of vehicle. Generally, the routes of the vehicles will not have any intersection (other than the depot) since the subsets of customer served by each vehicle are disjoint.

The parallel drone scheduling TSP (Murray and Chu, 2015)
A problem associated with devising optimal truck and UAV assignments in the case of a distribution center located in close proximity to customers (i.e., the UAV can fly directly from the DC to the customer and then come back) was first introduced in [43]. This work was the first to introduce a variant of the traditional TSP that address the challenge of determining optimal customer assignments for a UAV working in parallel with a delivery truck. This problem is defined as the parallel drone scheduling TSP and a mathematical programming formulation is provided. A heuristic to solve large scale instances is also described. It first assigns all the reachable customers to the drone and then the remaining ones to the truck. Then the algorithm reassign some drone customer to the truck and the truck route is modified until no possible savings are available. Finally, an extensive numerical analysis is conducted proving the effectiveness of the proposed heuristics and the benefits of last-mile parcel delivery by a UAV/truck parallel system.

Same-day delivery with drones and vehicles (Ulmer and Thomas, 2018)
The study presented in [59] analyzes the impact of using a combination of road-based vehicles and drone on the delivery costs and the customer served in Same-day delivery operations. To do so, they introduced a problem which considers a fleet of vehicles and a fleet of drones starting from a depot for servicing customers. The customers can make requests during a day that are unknown before the time of the order. Therefore, the provider has to decide whether or not an order can be served on the same day and whether a vehicle or a drone performs the delivery trying to maximize the expected number of customers served within the same day. The authors used an approximate dynamic programming technique known as a parametric policy function approximation to find a good decision policy for the provider. The parameter considered in the function is the travel time of the vehicle from the depot. This parameter represents a threshold that splits the service are in two zones. The customers that have vehicle time lower than the threshold should be preferably served by a drone while the others by the vehicles. The results showed that the customer partitioning obtained using the threshold increased the overall number of services.

A heuristic for the parallel drone scheduling TSP (Saleu et al., 2018)
An original solution method for the PDSTSP firstly defined in [43] was proposed in [39]. The authors showed that the PDSTSP can be considered as a bi-level program. In the first level, the customers are assigned to the vehicle or to the fleet of drones. In the second level, the route are determined, solving a TSP for the vehicle and a parallel machine scheduling problem for the fleet of drones. The proposed method is based on this consideration. Indeed, it starts from a sequence of all the customers that is splitted into a tour for the vehicle and series of trips for the drones. Then these routes are optimized and the whole method is repeated until the solution cannot be improved anymore. Moreover, they showed that if there is only one drone available then the proposed method is exact. Finally, computational results proved the effectiveness of the proposed approach compared to the heuristic proposed in [43].

Integrated scheduling of m-truck, m-drone, and m-depot (Ham, 2018)
A variant of the Parallel Drone Scheduling Problem is presented in [33]. The operating conditions of the considered problem are the following: multiple depots exists, from which multiple trucks and multiple drones must depart and return. A truck can serve multiple customers along its route but it cannot pickup any parcel. A drone can serve multiple customers and it can pickup parcels along its route. The capability of the drone to pickup parcels is considered to overcome the limited capacity characterizing this kind of vehicle. Customers can request multiple products each one with a different time-window. To better understand the difference between a traditional approach, where all the customers are served by truck, the system proposed in [43] and the proposed approach, the solutions obtained using the three systems are reported in Figure 2.6.a, Figure 2.6.b and Figure 2.6.c, respectively. In contrast to other approaches presented in literature the problem is tackled considering it as a parallel machine scheduling problem which minimizes the completion time with sequence-dependent setup (for travel-distance), precedence-relationship (for drop-pickup) and reentrant (for multi-visit and time-windows). Moreover, a constraint programming approach is proposed to solve the problem since this kind of approach excels most notably in scheduling application. Numerical results showed the efficacy of the proposed approach.



(a) The traditional approach, where a delivery truck visits all customer

(b) UAV/truck tandems proposed by Murray and Chu (2015)

(c) UAV/truck tandems with drop-pickup synchronization proposed in this paper

Figure 2.6: A comparison of the delivery schedule of the three systems (Source: Ham, 2018)

2.3.3 Vehicle Routing Problems with trucks and drones in tandem

In this subsection we report the contributions in which trucks and drones operate in tandem. In these problems, the required level of synchronization between the two vehicles is high. In addition to assignment and routing decisions there is another type of decision connected to the operations between trucks and drones in these problems. Indeed, for each drone sortie, the points where the truck launches and collects the drones have to be determined.

The flying sidekick traveling salesman problem (Murray and Chu, 2015)
The flying sidekick traveling salesman problem (FSTSP) was first defined in [43]. This work was the first to introduce a variant of the traditional TSP that address the challenge of determining optimal customer assignments for a UAV working in tandem with a delivery truck. The authors provided a mixed integer linear programming formulation that is able to solve instances of up to 10 customers. Therefore, they also proposed a route and re-assign heuristic. It first solves a TSP assigning the truck to visit all customers and then assign some customers to the UAV on the basis of the resulting saving. Finally, an extensive numerical analysis is conducted proving the effectiveness of the proposed heuristics and the benefits of last-mile parcel delivery by a UAV/truck tandem system.

Optimization of a truck-drone delivery network (Ferrandez et al., 2016)
The work presented in [31] has a twofold objective. On one hand it investigates the time and energy associated to a truck-drone delivery system compared to a standalone truck or drone system. On the other hand it proposes an optimization algorithm that determines the number of launch sites and locations and the optimal total time on the basis of the algorithm result. The considered problem assumes that one or more drones and a single truck work in tandem to deliver packages. The drones are not constrained by range to gain a better sens of the upper/lower boundaries of time and energy. However, the truck has to wait stationary while a drone is airborne. The proposed algorithm first determines k truck stops where the drone will be launched using the well-known $k - means$ algorithm. Then, it determines the best truck route between those launch locations using a genetic algorithm. Computational experiments showed the energy expenditure and the completion time for different values of the number of launch locations.

Coordinated Truck-and-Drone Logistics (Carlsson and Song, 2018)
An analysis of the efficiency of a hybrid approach in which a UAV provides service to customers while making return trips to a truck that is, itself, moving is presented in [13]. In other words, a UAV picks up a package from the truck (which continue on its route), and after delivering the package, the UAV returns to the truck to pick up the next package. The analysis goal is to reduce the problem to a small set of parameters, using the continuous approximation paradigm, and then determine how these parameters affect the outcome of the problem. The parameters identified by the authors were the truck and drone speeds and the probability distribution function of the customers on a compact region. On the basis of the asymptotic theoretical analysis performed, the authors concluded that the improvement in the efficiency due to augmenting a delivery truck with a UAV is related to the square root of the ratio of the speeds of the truck and the UAV.

Worst-case analysis for the VRP with drones (Wang et al., 2017)
The first formal description and definition of the Vehicle Routing Problem with Drones (VRPD) is given in [62]. The authors proved several worst case theorems and their goal is to provide theoretical bounds on the benefit from using drones. In each of the theorems presented, two related problems are compared. These problems have the same set of customers but served with a different fleet. In the first one, the fleet consists of trucks only while in the

second one of trucks and drones. Each result reveals the amount of time that could be saved, in the best case, as a result of using trucks and drones rather than trucks alone in delivering packages to customers.

Models and Connections for the VRP with drones (Poikonen et al., 2017)
The same authors of [62] proposed an extension of their work in [48]. They first summarize the key results of the previous paper since they will serve as templates for further proofs. Then, they extend some worst-case bounds to more generic distance/cost metrics. Moreover, they also consider limited battery life and cost (in addition to the completion time) objectives. Finally, the authors highlight the connection between the VRPD and two well-known problems in literature as the min-max close-enough VRP (CEVRP) and the min-max VRP, where the objective function of both problems is the minimization of the longest route. In particular, they showed that the VRPD objective value is bounded below by the CEVRP objective value and above by the objective value of the VRP. The authors underline that other than the bounds on optimal objective values no relationship is known between the optimal solutions of these problems.

Vehicle Routing Problems for Drone Delivery (Dorling et al., 2017)
A problem in which the delivery fleet consists of a fleet of drones is presented in [29]. The authors defined this problem as the drone delivery problem and they tackled it as a multitrip VRP. Indeed, compared to the classical setting of the VRP, the multitrip ability compensates the limited drone payload by reusing drones when possible. Moreover, the MTVRP developed considers battery and payload weight when calculating energy consumption. In addition to the problem definition, the authors proposed a linear energy consumption model for multirotor drones. On the basis of this consumption model, the authors formulated the MTVRP as mixed integer linear programming model. In particular, two version of the problem are given. In the first one is minimized the cost of deliveries while the overall delivery time is minimized in the second one. Finally, a string-based simulated annealing algorithm is proposed for solving the MTVRP on practical scenarios with hundred of locations. Numerical results showed that optimizing the battery weight resulted in a saving of 10% compared to the case where the battery weight is the same for each drone. On the other hand, a saving of up to 80% is obtained optimizing the payload weight.

Scheduling truck and autonomous robot deliveries (Boysen et al., 2018)
A new truck-based robot delivery concept is presented in [10] on the basis of the strategic partnership between Mercedes-Benz Vans and a start-up company which develops autonomous robots for last-mile deliveries. In the considered system, a truck loads the parcels for a set of customers at a central depot. A fixed part of the truck capacity is reserved for the autonomous robots on board. The truck moves into the city center and launches one or more robots to deliver the parcels once it reaches a drop-off point. The robots have single capacity and after the delivery they come back to a decentralized robot depot within the city center. Then, the truck moves onwards to successive drop-off points until all robots are launched. If the truck has to serve other customers and no robots are available on board then it can move to a decentralized robot depots to load other robots. This process is repeated until all the customers are serviced. The aim of the problem is to determine the truck and robots routes such that the number of late customer deliveries is minimized. The authors defined this problem as the truck-based robot delivery scheduling problem and they formulated it through Mixed Integer Programming model. Moreover, they develop an efficient approach to determine the optimal assignment of customers to drop-off points and robot depots given a fixed truck route. On the basis of this approach, they proposed a multi-start local search that at each iteration determines a new truck route that represents the input of the approach for the optimal assignment of the customers. Computational study showed that the decentralized robot depots contribute to an efficient delivery process since they avoid the waiting times that arise in the case the truck has to wait for the return of the robots.

A decomposition-based algorithm for TSP-D (Yurek and Ozmutlu, 2018)
An improved mathematical model for the TSP-D is presented in [63]. The proposed model is not able to solve instances with more than 10 customers within one-hour computational time limit. To overcome this limit, the authors proposed an algorithm in which a mathematical integer programming formulation is iteratively solved. This algorithm decomposes the problem into two stages. The truck route and the drone route are determined in the first and second stage, respectively. In the first stage, the customer assignment is also determined since the customer that are not in the truck route will be served by the drone. The drone route is then defined in the second stage solving a MIP formulation that determines for each customer that is serviced by the drone its launching and landing locations. The proposed algorithm was tested using

randomly generated instances and compared with the solutions of different formulations proposed in other papers. Experiments showed that the proposed moethod was able to solve instances up to 12 customers.

A metaheuristic for the FSTSP (De Freitas and Vaz Penna, 2018)
The effectiveness of the combination of an UAV and a truck for last mile parcel delivery is analyzed in [25]. A hybrid local search heuristic based on the Randomized Variable Neighborhood Descent (RVND) method is proposed. The proposed heuristic is composed of three steps. In the first step, the optimal TSP solution, i.e., when all customers are served by the truck is generated using a TSP solver. In the second step, the TSP solution is modified removing some customers from the truck route and determining how to serve them with the UAV. Finally, in the last step, the solution is optimized applying the RVND method. The authors proposed a new set of benchmark instances generated from the TSPlib. Computational experiments showed that the potential reduction of the total delivery time can be up to 20%, compared to a route without the drone.

Algorithms for the VRP with drones (Schermer et al., 2018)
Two heuristics for solving large scale VRPD are proposed in [52]. The first heuristic, called Two-Phase Heuristic, initially ignores the drones and focuses on constructing good VRP tours. Then, starting from the first vertex in a tour, all the customers are analyzed and whenever a feasible sortie is possible and reduce reduces the time required to complete the tour, the drone is inserted. The solution so obtained is optimized trying to exchange the method of delivery of two customers that are serviced by a truck and a drone, respectively. The second heuristic, called Single-Phase Heuristic, inserts drones from the beginning when the routes are determined. The performance of the two heuristics were evaluated experimenting them on large-scale TSP instances adapted for being used in the context of the VRPD.

Optimization Approaches for the TSP-D (Agatz et al., 2018)
The traveling salesman problem with drone (TSP-D) was first defined in [1]. The main contributions of the paper are the following: 1) a new IP model is developed that is able to solve instances of up to 12 nodes; 2) A greedy partitioning heuristic and an exact partitioning algorithm based on dynamic programming, to perform the assignment of truck and drone deliveries for any given delivery sequence, are described. The dynamic programming algorithm

find the optimal assignment and it is based on a route first cluster second procedure. It first constructs a tsp solution using a TSP solver. Then, the TSP-D solution is constructed assigning some nodes as drone nodes and some as truck nodes; 3) a theoretical analysis of the worst case approximation guarantee of the heuristic is conducted; 4) a new set of benchmark instances is generated; 5) a numerical study of the truck and drone delivery system is conducted to evaluate its performance on the basis of different customer densities, geographical distributions, and drone speeds. These results showed that substantial savings are possible using a combined truck and drone system compared to the truck-only solution.

Dynamic programming approaches for the TSP-D (Bouman et al., 2018)
Exact solutions approaches for the TSP-D based on dynamic programming are presented in [8]. The authors first introduced a 3-pass dynamic programming approach and then they extend the last pass of this approach to an A* algorithm (i.e., a graph traversal and path search algorithm). The three passes of the approach are the following: 1) enumerate the shortest paths for the truck for every start node, end node, and set of truck nodes covered by the path; 2) combine these truck paths with drone nodes to obtain efficient operations (i.e., operations that represents the least costly way to cover a set of nodes with an operation); 3) compute the optimal sequence of these operations such that all locations are covered and the sequence start and ends at the depot. Computational experiments showed that the proposed dynamic programming approaches were able to solve larger problems (up to 20 nodes) than those solved by the mathematical programming approach presented in literature so far.

On the min-cost Traveling Salesman Problem with Drone (Ha et al., 2018)
A variant of the TSP is presented in [32]. This variant is called min-cost TSP-D, the objective is to minimize the total operational cost of the system including two distinguished parts. The first part is the transportation cost of truck and drone while the second part relates to the waste time a vehicle has to wait for the other whenever drone is launched. The authors proposed a new MILP model whose main difference with the formulation presented in [43] is that the waiting time is captured by a variable in order to calculate the waiting cost of the two vehicles. Moreover, two heuristics to tackle the problem are described. The first one is a greedy randomized adaptive search procedure (GRASP) that, at each iteration, first constructs a TSP and then generates a

TSP-D solution splitting the customers between drone and truck. Instead, the second heuristic is based on the heuristic proposed in [43]. In particular, the authors modified the calculation of the cost in the heuristic replacing the times with the cost and adding a component that takes into account the waiting times. The computational results showed that the GRASP heuristic outperforms the second one.

Drone scheduling for given truck routes (Boysen et al., 2018)
The problem tackled in [9] address the truck-based drone delivery problem from a different perspective. Indeed, the authors assume that the truck route is already given and the objective is to optimize the schedule of drones launched from the truck when servicing a given set of customers. They assume that drones can leave the truck and return to the truck only at predetermined stops of the truck long its route. On this basis, six variants of the drone schedule sub-problem are investigated. Indeed, they considered single or multiple drones based on the truck and three degree of freedom with respect to where a drone returns to the truck. In particular, the three cases are the following: 1) the start stop and return stop of a drone trip are the same; 2) the return stop is the immediately successive stop after the start stop; 3) the return stop can be one of the successive stop after the start stop. Two mixed integer programming model are presented and can be adapted to tackle the six presented variants of the considered problems. As the two formulations presented in [16], the main difference between these two formulation is represented by the third index used to have the information about the route of each drone that it is not present in the second formulation. Furthermore, they showed how to integrate the drone sub-problem into a metaheuristic framework, if the original problem requires the determination of the truck route.

FSTSP with Payload dependency and No Fly zones (Jeong et al., 2019)
A variant of the FSTSP which considers no-fly zones during certain period and energy consumption on the basis of the weight carried by the drone is presented in [35]. The shape of the no-fly zone is a circle and the zones do not overlap each other. Moreover, these areas can forbid drone operation only at a specific time or indefinitely. Instead, the possible flight duration is estimated considering a model which uses the loading weight to compute energy consumption. These new characteristics of the problem are taken into account in a mathematical formulation that is proposed in the paper. Computational results showed that a commercial MIP solver was able to solve the proposed

model on instance of up to 10 customers in several hours. Therefore, the authors proposed a two phase construction and search algorithm to tackle larger scale problems. In the first phase, an initial truck routes is generated. Then, it proceeds to assign some customers to the drone simultaneously determining the corresponding routes. After having determined a total route, in the second phase, a section of the truck route is optimized by a local search trying to improve the truck and drone synchronization. The results showed that considering no-fly zones and the package weight in the energy expenditure reduce the advantages of drone operations in a truck and drone delivery problem since the use of drones is limited.

The hybrid vehicle-drone routing problem (Karak and Abdelghany, 2019)
An integrated vehicle-drone system is presented in [36] and defined as the mothership system. This system consists of vehicles that carry drones from depots to locations where the drones are dispatched to perform multiple pick-up and delivery operations. The mothership system allows dozens of pick-up and deliveries simoultaneously since the UAV are dispatched according to a swarm-like approach. The authors proposed a mixed-integer programming formulation that determines the routes of the vehicle and of the drones minimizing the total cost of the pick-up and delivery operations. The optimal solution of the formulation can be obtained in a reasonable computation time for small problem. Therefore, the authors developed a method based on the well-know Clarke and Wright heuristic and called hybrid Clarke and Wright heuristic. This heuristic considers at the same time the saving costs for both the vehicle and the drones so generating an efficient multimodal delivery system. It is compared with two other heuristics proposed in the same paper and called vehicle-driven heuristic and drone-driven heuristic, respectively. The vehicle-driven heuristic first determines the vehicle route and then the drone routes while the drone-driven use the reverse approach. Computational results showed that the first heuristic outperforms the other two since they mainly optimize the cost of one delivery mode only.

On a truck-and-drone delivery system (Crişan and Nechita, 2019)
A new heuristic for solving the FSTSP is proposed in [20]. This heuristic first solve a TSP considering all the customers served by the truck and then constructs in a greedy way the drone route. In particular, it removes from the truck path the nodes that result in the biggest savings from the truck route length point of view. The customer visit order determined by the TSP

solution is not changed while designing the drone route. This assumption makes this approach computationally efficient since its major advantage is its speed. The heuristic is tested on two instances generated in the Romanian (with 2950 customers) and Bulgarian region (with 1954 customers), respectively. Despite the large size of the instances, the computational results proved the speed of the approach since they are solved within 90 seconds.

Parcel delivery by vehicle and drone (El-Adle et al., 2019)
A new mathematical formulation for the TSP-D is presented in [30]. The model is enhanced by employing cut generation and bound improvement strategies. In particular, the cut generation is based on the introduction of a series inequalities which ensure the connectivity of the subgraph induced by the movement of the two vehicles. The bound improvement, on the other hand, is obtained by defining a set of M parameters, each tuned to satisfy a specific constraint in the proposed model instead of have a single M throughout the formulation. These values are determined on the basis of the duration of a feasible TSP-D solution. Therefore, the authors also proposed a greedy heuristic. The heuristic determines a solution starting at the last node visited (the depot at the first iteration) and analyzing its first two nearest nodes. If it is convenient to serve the first nearest node with the drone and then recollect it in the second nearest node location then the algorithm assigns a drone launch to the first nearest node. Otherwise, the first nearest is served by the truck with the drone on board and the process is repeated until every customer is served. The results showed the effectiveness of the new formulation since instances up to 24 nodes are solved to optimality.

A metaheuristic for the VRP with drones (Sacramento et al., 2019)
An extension of the VRP where each truck collaborates with a single UAV is studied in [50]. The problem is a variant of the FSTSP for the multi-truck case, and includes capacity and time completion constraints, while having cost minimization as objective function. The capacity constraint is devoted to take into account the capacity of the truck. The time completion constraints, on the other hand, ensure that driver maximum workable-hours per day is respected. Contrary to the FSTSP, the objective function is focused on the route total cost rather than on reducing the completion time. The authors presented a new mathematical formulation for the problem and an adaptive large neighborhood search metaheuristic. The metaheuristic is based on the use of a set of repair and destroy methods. At each iteration a destroy method is randomly selected

and it is applied to the current solution. Then the solution is repaired using a randomly selected repair method. The repair methods ensure that an infeasible solution cannot be obtained. Computational results showed the performance of the algorithm and the differences with the VRP using different parameters for the problem (i.e., drone endurance, drone speed, payload capacity, etc.) and proved the clear advantege of using drones for delivery activities.

The VRPD with en route operations (Schemer et al., 2019)
An extension of the VRP with drones is proposed in [53] and it is called the VRP with Drones and En Route Operations. This new problem assumes that drones might also be launched and retrieved at some discrete locations on each arc. The authors formulated the problem as a mixed integer linear programming model that can be used to solve small-scale problem within a reasonable computational time. To tackle larger instances, the authors proposed a heuristic that combines elements from Variable Neighborhood Search and Tabu Search metaheuristics. The structure of the algorithm is similar to the basic VNS (initialization, shaking step and local search). In particular, the shaking step is used to determine different truck routes while the local search inserts the drone operation. A tabu list, which stores the truck routes, is implemented in the algorithm to avoid that a given truck route is optimized by the local search multiple times . Numerical results showed that en route operations reduce the completion time and increase the drone utilization. These advantages are notably greater when the drone endurance is small or the drone speed is relatively higher compared to the truck speed.

Multi-visit drone routing problem (Poikonen and Golden, 2019)
The multi-visit drone routing problem was first defined in [47] and it considers a tandem between a truck and a drone for servicing a set of customers. The proposed problem allows for a drone to carry multiple heterogeneous packages but also allows a specification of the energy drain function that considers each package weight. Each drone can serve one or more customers and it may return to the truck, acting as a mobile depot, to swap/recharge its battery and pick up a new set of packages. A linear integer programming model based on the concept of operation is formulated to define the problem. An operation is a set of actions beginning with the truck and the drone at a launch location and terminating with both the vehicles in a retrieving location. During an operation, the drone may launch from the truck, visit one or more customer and then rendezvous with the truck that has to travel

directly to the retrieving location after having launched a drone. A heuristic solution approach is proposed since the number of feasible operations may be extremely large as the size of the instance grows. This approach is based on a graph transformation that given a fixed customer visit order transforms the original problem in a shortest path problem. Computational experiments showed that the completion time is highly sensitive to drone speed.

New formulations for the FSTSP (Dell'Amico et al., 2019)
A study tackling the flying sidekick traveling salesman problem is reported in [26]. The authors focus their attention on the mathematical formulation of the problem. They first improve the formulation proposed in [43]. Then, they propose two original formulations substituting some explicit constraints with exponentially many constraints that will be added in cutting plane fashion. Indeed, the two formulations are solved by a Branch-and-Cut algorithm. The main difference between the two original formulations is represented by the number of indexes used for representing the variable a drone sortie. In particular, the first one uses three indexes (launch location, customer served, retrieval location) while the second one uses only two indexes (start location, end location). Moreover, a set of valid inequalities are proposed. Computational results were carried out on literature instances ([43] and [63]). They showed that the 2-index formulation outperfomed the other two tested formulations.

2.4 A classification scheme for drone management problems in logistics

In this section, a classification of the papers examined in the previous section is provided. In particular, Table 2.2 extends the table proposed in [16] in terms of parameters and analyzed papers. The first two columns report the year and the authors of the paper. The third and fourth columns show the number of trucks and the number of drones considered in the problem tackled by each paper. In particular, if trucks are used the number of drones is related to the drones available on each truck, otherwise it refers to the drones present at the depot. The fifth column reports the number of customer served by a drone during a single sortie. This column is filled with 1 if, for technical reason, it is assumed that a drone cannot serve more than one customer and with k if the number of customer served per sortie depends on different factors (e.g., distance travelled, package weight, etc.). and so it is not set an apriori upper limit. The successive column indicates if there is synchronization between the two kind of vehicles. If *Yes* is reported, then it means that the two vehicles need to be synchronized since the truck has to launch and to recollect the drone. If *No* is reported, then it indicates that the two vehicles serve the customers independently. If no truck is considered in the tackled problem, then a dash (-) is reported. The column *solution approach* indicates the method used by the authors in tackling the studied problem. The column named *reference problem* reports the problem indicated as the reference problem in the paper. The next column shows the largest instance size solved in the each paper. The value reported indicates the number of customers served in the largest instance. If no numerical test was performed, then the term *N/A* is reported. The column *Notes* gives further information about the problem settings and the instance used in the computational experiments. In filling this column, two assumptions were made: the standard objective is the completion time minimization (or a function of it) of the delivery task; the truck is allowed to served customer, if the truck is considered in the problem. These assumptions allow us to reduce the information in the table. Indeed, we explicitly report details about the truck and the objective function only if they are different from these assumptions. Finally, in the last column is reported an original classification for the problem with drones. In particular, the proposed notation is similar to the one proposed for scheduling problems in [15]. The format of the proposed notation is based on three parameter *X-Y-Z*, where:

- X represents the type of coordination between the truck and the drone. It can assume two values: P if the two vehicles work in parallel since they act independently; T if the two vehicles work in tandem.

- Y indicates the number of trucks available. It can assume three possible values: N if no truck are used in the considered problems (this means that the deliveries will be performed only by drones); S (M) if a single (multiple) truck(s) is (are) available. If this symbol is followed by a star (*), then it means that the truck cannot perform any delivery.

- Z indicates the number of drones available. If trucks are used it will refer to the number of drones on each truck, otherwise it will refer to the number of drones available at the depot. It can assume two values: S (M) if a single (multiple) drone(s) is (are) available. If a drone can perform multiple deliveries per sortie, then the symbol will be preceded by a k.

Year	Authors	Truck	Drone	Customers per sortie	Synchr.	Solution approach	Reference Problem	Instance size	Notes	Classification
2015	Murray and Chu (1)	Single	Single	1	Yes	• MIP formulation • Heuristic algorithm	TSP	10	• First paper to define the hybrid truck and drone delivery problem • Generation of benchmark instances	T-S-S
2015	Murray and Chu (2)	Single	Multiple	1	No	• MIP formulation • Heuristic algorithm	Scheduling	20	• First paper to define the parallel truck and drone delivery problem • Generation of benchmark instances	P-S-M
2016	Ferrandez et al.	Single	Multiple	1	Yes	• Newthon's method • k-means based algorithm	TSP	250	• The truck cannot serve any customer while a drone is dispatched	T-S*-M
2017	Ulmer and Thomas	Multiple	Multiple	1	No	• Policy function approximation approach • Markov decision process model	VRP	800	• Maximization of the customer served within the time horizon • Customers dynamically request delivery • Requests are stochastic	P-M-M
2017	Carlsson and Song	Single	Single	1	Yes	• Continuous approximation analysis	TSP	100	• Theoretical analysis performed in the Euclidean plane	T-S-S
2017	Wang et al.	Multiple	Single	1	Yes	• Worst case analysis	VRP	N/A	• First paper to define the hybrid truck and drone delivery problem with multiple trucks	T-M-S
2017	Poikonen et al.	Multiple	Single	1	Yes	• Worst case analysis	VRP	N/A	• Connection between the objective values of the VRPD, CEVRP and VRP	T-M-S
2017	Dorling et al.	Multiple	Single	k	Yes	• MILP formulation • Simulated annealing algorithm	VRP	500	• Implementation of Battery weight and Payload energy dependency	T-M-kM
2017	Coelho et al.	None	Multiple	k	-	• MILP formulation • Metaheuristic algorithm	VRP	25	• Multi-objective function • Implementation of recharge station	P-N-kM
2018	Saleu et al.	Single	Multiple	1	No	• MILP formulation • 2 Step heuristic	Scheduling/TSP	229	• Instances from TSPLIB and Murray and Chu's paper	P-S-M
2018	Ham	Multiple	Multiple	k	No	• Constraint programming approach	Scheduling	100	• Implementation of pick-up tasks for drones • Multiple depots hosting a fleet of vehicles and drones are available	P-M-kM
2018	Boysen et al.	Single	Multiple	1	Yes	• MIP formulation • Heuristic algorithm	TSP	40	• Minimization of the number of the late deliveries • The truck is not able to serve any customer • Implementation of stations where vehicle and drones may wait for each other • It is not mandatory to recollect drones	T-S-M
2018	Yourek and Ozmutlu	Single	Single	1	Yes	• Iterative 2 phase heuristic algorithm	TSP	20	• Instances from Agatz et al.	T-S-S
2018	De Freitas and Vaz Penna	Single	Single	1	Yes	• 3 steps randomized variable neighborhood descent heuristic algorithm	TSP	100	• Instances from TSPLIB	T-S-S
2018	Schermer et al.	Multiple	Single	1	Yes	• Heuristic algorithms	VRP	100	• Instances from TSPLIB	T-M-S
2018	Bouman et al.	Single	Single	1	Yes	• Dynamic Programming Approach • Heuristic algorithm	TSP	10	• Instances from Agatz et al.	T-S-S
2018	Agatz et al.	Single	Single	1	Yes	• MIP formulation • Dynamic Programming Approach • Heuristic algorithm	TSP	10	• Generation of benchmark instances	T-S-S

Table 2.1: (First page) Classification of the OR contributions on logistics with drones

Year	Authors	Truck	Drone	Customers per sortie	Synchr.	Solution approach	Reference Problem	Instance size	Notes	Classification
2018	Ha et al.	Single	Single	1	Yes	• MIP formulation • Heuristic algorithms	TSP	100	• Minimization of costs related to waiting times and travelled distances	T-S-S
2018	Cheng et al.	None	Multiple	1	Yes	• MILP formulation • Branch-and-Cut algorithm	VRP	40	• Implementation of Payload energy dependency and time windows • Extensive literature review provided	P-N-M
2018	Boysen et al.	Single	Multiple	1	Yes	• MIP formulation	TSP	100	• The problem tackled considers fixed the truck route	T-S-M
2019	Shavarani et al.	None	Multiple	1	No	• Fuzzy MILP formulation • Genetic Algorithm	Faciliy location	14304	• Minimization of the number of warehouses and drone travel distances • Use of refuel station to extend drone range	P-N-M
2019	Aurambot et al.	None	Multiple	1	No	• Greedy heuristic	Faciliy location	N/A	• Instances generated on the basis of real data	P-N-M
2019	Jeong et al.	Single	Single	1	Yes	• MILP formulation • 2 Phase heuristic	TSP	50	• Implementation of Payload energy dependency and no fly zones • Instances from Murray and Chu's paper	T-S-S
2019	Ulmer and Thomas	None	Multiple	k	No	• Policy function approximation approach • Markov decision process model	VRP	1000	• Customers dynamically request delivery • Requests are stochastic • Implementation of pick-up stations	P-N-kM
2019	Liu	None	Multiple	1	No	• MIP formulation • Heuristic algorithm	VRP	533	• Minimization of the delivery lateness • Customers dynamically request delivery • Implementation of recharge stations	P-N-M
2019	Karak and Abdelghany	Single	Multiple	k	Yes	• MIP formulation • Heuristic algorithm	VRP	100	• Implementation of pick-up tasks for drones • The truck is not able to service any customer • Implementation of stations where vehicle and drones may wait for each other	T-S*-kM
2019	Crisan and Nechita	Single	Single	1	Yes	• Heuristic algorithm	TSP	2950	• None	T-S-S
2019	El-Adle	Single	Single	1	Yes	• MIP formulation	TSP	32	• Instances from Agatz et al. • Implementation of valid inequalities and scaled Big M values	T-S-S
2019	Sacramento	Multiple	Single	1	Yes	• MIP formulation • Adaptive large scale neighborhood search matheuristic	VRP	200	• Implementation of different destroy and repair methods	T-M-S
2019	Schermer et al.	Multiple	Single	1	Yes	• MILP formulation • Hybrid VNS/Tabu search algorithm	VRP	10	• Drones can be launched and retrieved along the edges	T-M-S
2019	Poikonen and Golden	Single	Multiple	k	Yes	• MIP formulation • Heuristic algorithm	VRP	100	• Implementation of Payload energy dependency	T-S-kM
2019	Dell'Amico et al	Single	Single	1	Yes	• MIP formulation	TSP	13	• Instances from Murray and Chu	T-S-S

Table 2.2: (Continued from previous page) Classification of the OR contributions on logistics with drones

Based on the literature and the related classification, it is possible to define other three main categories for the papers on the basis of the number of truck considered:

- A: No truck and multiple drones (6 papers)
- B: Single truck (18 papers, 11 of them consider a single drone and 7 of them consider multiple drones)
- C: Multiple trucks (8 papers, 6 of them consider a single drone on each truck and 2 of them consider multiple drones on each truck)

To summarize, the papers related to each class are: A - [19, 16, 55, 3, 58, 38]; B - [41, 31, 13, 39, 10, 63, 25, 8, 1, 32, 9, 35, 36, 20, 30, 47, 26]; C - [59, 62, 48, 29, 33, 52, 53].

Furthermore, the following observations can be made:

1. the truck-drone tandem system (B and C categories) is the most intensively studied problem;
2. almost all surveyed studies assume that drone flight range depends exclusively on the distance travelled;
3. most papers addressing the truck-drone system assume that during each drone sortie it can serve at most one customer;
4. most papers addressing the truck-drone system assume that the truck can serve customer while the drone is detached;
5. most studies on the truck-drone system assume that there is only one drone on each truck (this is due to the complexity arising from the required synchronization between truck and drone).

2.5 Conclusions

An extensive and detailed review of the scientific literature on optimization problems involving the use and management of drones has been reported in this section. The review is focused on problems arising in surveillance and logistics sectors. In particular, we deeply analyzed the problems arising from logistics and propose a new classification based on the kinds of vehicles involved and on the synchronization level between the vehicles. On the basis of this classification, the truck-drone tandem system resulted the most intensively studied problem. Therefore, the studies presented in the successive chapters focused on different variants of this problem.

Chapter 3

Transformation of Truck-and-Drone Coordination Problems into Traveling Salesman Problems

3.1 Introduction

The number of literature contributions which considers civilian uses of drones, among which exists a class of problems that may be described as truck-and-drone coordination problems, is rapidly increasing. Problem definitions, model assumptions, and even the kind of vehicles used differ, but they generally have a premise in common.

In particular, in a truck-and-drone coordination problem there exists a main carrier vehicle that is capable of deploying one or more generally smaller vehicles. The combination of the carrier and deployed vehicles must serve multiple locations, typically minimizing the completion time or the total operational cost. Each deployed vehicle must repeatedly come back to the carrier vehicle, typically to pick up new cargo, share gathered information, or replenish its battery.

The majority of the contributions on the topic explicitly considers the case of same-day parcel delivery, where the main carrier vehicle is a truck and the smaller deployed vehicle(s) is an unmanned aerial vehicle (drone) or a ground-based robot (droid). For simplicity, we henceforth call the carrier vehicle the truck and any deployable vehicle will be called a drone. We will call this class of problems truck-and-drone coordination problems.

In this chapter, we will display that a broad swath of truck-and-drone coordination problems can be recast as an ordinary traveling salesman problem (TSP). To solve the resulting TSP, we will exploit the capabilities of existing solution techniques for the TSP.

3.2 Related works

As already highlighted in the first chapter, a recent survey article [45] explores the many uses of drones in a civilian context. Application areas of drones include agriculture, glaciology, mapping, target tracking, entertainment, security, and infrastructure inspection, among others. However, one specific area of focus that has garnered significant attention involves the use of drones in the context of consumer parcel delivery, especially last-mile parcel delivery.

In particular, mainly due to the limited range and payload capacity of drones, several papers have considered hybrid models of delivery which require synchronized behavior between one or more trucks and one or more drones. First among these [43] introduced the Flying Sidekick Traveling Salesman Problem (FSTSP). In this problem, a single drone, capable of serving one customer per sortie, is allowed to launch from the truck to assist it in delivery. It may return to the truck to replenish its battery and pick up further parcel. The drone battery may represent a maximum flight duration constraint, and while the drone is airborne the truck may make deliveries. The objective is to minimize the elapsed time until both vehicles have arrived back to the depot. Murray and Chu present both a formulation and heuristic solution methods. A similar problem has been studied [1], titled the traveling salesman problem with drone (TSP-D), which differs slightly from the FSTSP because it allows the launch and landing site for a drone to be the same location. The authors propose a mixed integer linear programming (MILP) formulation and a family of heuristics. Each of these heuristics is based on a framework that may be described as "route first, partition second", where the order of customer delivery is determined in the first step, and the partitioning of the customer set into truck-delivered and drone-delivered package sets is determined in the second step. Another paper [32] studies the min-cost (rather than min-time) version of the TSP-D, which accounts for additional costs that may occur in drone operations (e.g., costs connected to the waiting times). Their work proposes two heuristics: a greedy randomized adaptive search procedure (GRASP) heuristic and a heuristic called TSP-LS, which extends a heuristic proposed by Murray and Chu. Yurek and Ozmutlu [63] describe an iterative decomposition algorithm to solve the TSP-D. Freitas and Penna [25] propose a variable neighborhood descent heuristic to solve the FSTSP.

The Horsefly Problem [13] has also been studied where a truck-and-drone tandem operate in the Euclidean plane; the main difference with the other studies consists in the assumption of a non-discrete set for the set of launch locations. The drone has unit capacity and the goal is the minimize route completion time. Asymptotic analyses are conducted and bounds are established on objective values as the customer density moves toward infinity. In another paper [46], a similar problem is studied, which the authors refer to as the Mothership and Drone Routing Problem. They propose a branch-and-bound tree to explore potential visit orders and a second order cone program to optimally select launch and rendezvous locations relative to a fixed visit order. In Campbell et al. [11], continuous approximation methods are used and the potential economic benefit of drones under a variety of assumptions (e.g., differing numbers of drones and customer densities) is assessed. A different model of truck-and-drone coordination has been considered [24], where drones fly from the depot to resupply trucks that are on their delivery routes.

The Vehicle Routing Problem with Drones [62] contains multiple homogeneous trucks, each capable of launching one or more drones. Several theoretical bounds regarding maximum speed-up ratios are established. The same authors extend this work [48] and draw a relationship to the close-enough traveling salesman problem.

As the current chapter will eventually propose a transformation into a TSP, we would like to note the classic work [37] that described an extremely efficient heuristic approach for solving the TSP, which still provides great utility today, including in more recent implementations such as LKH2 [34].

3.3 Preliminaries

Let V be a discrete set of locations where a drone is allowed to launch from or land on a truck.[1] Let C be a set of customer locations that must be serviced by the truck or a drone. The problem objective is to service all customers with feasible operations in a least cost manner.

Let us define an *operation* and related terms. An operation o is a set of actions beginning at a location known as the *launch point* of the operation. We denote the launch point of operation o as $l(o) \in V$. An operation ends at a location called the *rendezvous point* of the operation, denoted $r(o) \in V$. The truck and all drones must be present at the launch point at the start of the operation. The vehicles may then disperse, potentially service one or more customers, and eventually regather at the rendezvous point of the operation. If the problem considers multiple drones for a truck, then the launch/rendezvous point (i.e., the beginning and the end) of an operation is the point where all the vehicles (the truck and the multiple drones) are present together. Drones that are not servicing customers may ride atop the truck instead. In Figure 3.1 is reported an example of a route which contains two operations o_1 and o_2. Green dashed segments show the flight path of drones during an operation. Black dashed segments show the path of the truck during an operation. Black solid segments show the path of the truck relocating (deadheading) between consecutive operations. The red squares are customer locations. Blue circles are either launch points or retrieval points for operations. The four red squares towards the top left, collectively, are the cover of operation o_1. The two red squares bottom right, collectively, are the cover of o_2. A feasible route is a closed tour that begins and terminates at the depot, and consists of an alternating sequence of feasible operations and truck relocation.

The set of customers that are serviced by the truck or any drone during an operation o is denoted $cov(o)$. We denote the *size* of the operation as the number of customers in its cover: $s(o) = |cov(o)|$. We require that, within each operation, at least one vehicle among the truck and drone(s) services one or more customer locations. That is, $s(o) \geq 1$.

[1]The model easily extends to the case where we separately define V^- to be a discrete set of locations where a drone may launch from a truck and V^+ to be a discrete set of locations where the drone may return to the truck. However, for ease of notation we assume that $V = V^+ = V^-$.

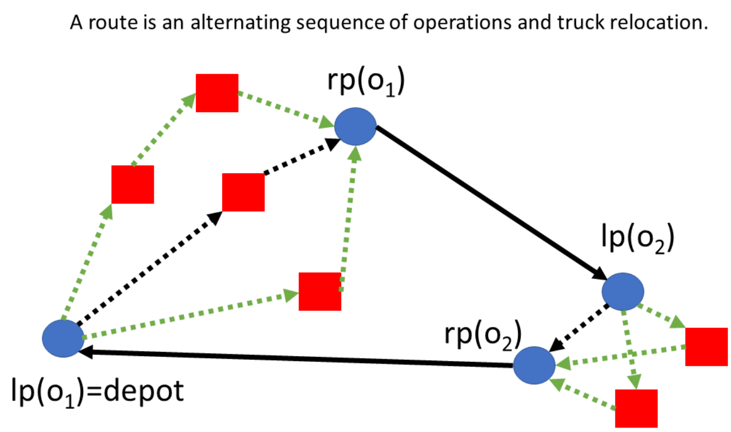


Figure 3.1: An example route containing two operations.

The feasibility of an operation o is denoted $feas(o)$; the cost associated with an operation is denoted $cost(o)$.

In the following, we will assume that for any prospective operation o, we know or can easily compute in advance:

- $l(o)$
- $r(o)$
- $cov(o)$
- $feas(o)$
- $cost(o)$

Each of the five elements above are assumed to be problem inputs. In particular, we highlight that the feasibility and cost of an operation should be defined with respect to the constraints and objective function for the specific problem under study. On the basis of the addressed problem, the computation of the cost and/or feasibility of each operation may be a simple calculation, or potentially a complex subproblem, depending on the related assumptions.

Moreover we make a few additional assumptions:

- the cost and/or feasibility of one operation have no impact on the cost and/or feasibility of another operation (we will refer to this property the *separability* of operations).
- the cost of any operation is non-negative.
- for two feasible operations o_1, o_2 where $l(o_1) = l(o_2)$ and $r(o_1) = r(o_2)$ and $cov(o_2) \subseteq cov(o_1)$, if o_1 is feasible, then o_2 must also be feasible, and $cost(o_2) \leq cost(o_1)$. In other words, the addition of extra customers to an infeasible operation cannot make it feasible, and the addition of a customer to an operation cannot lower the cost of the operation.

Finally, any feasible route must service all customers, and the objective is defined as minimizing the total cost, which is the simple sum of the costs of all operations that are chosen as part of the route plus the costs associated with relocating the truck and drones between consecutive operations.

3.4 Solution Method

In this section, each phase of the proposed solution method will be described in detail. Each operation is split in multiple copies, one for each customer served, so generating the set of split operation in the first phase. In the second phase, the construction of an Equality Generalized TSP is performed. Finally, in the third phase the resulting E-GTSP is transformed into an Asymmetric TSP.

3.4.1 Formation of the Set of Split Operations

Enumerate All Operations

In the first phase of the solution method, the first step is the generation the set of all feasible operations, denoted $F(O)$. For each $o \in F(O)$, we should retain the following information: $l(o), r(o), cov(o), cost(o)$.

Removal of Dominated Operations

Once the set of feasible operations is generated, we prune this set removing the dominated operation. In particular, for any pair of operations $o_1, o_2 \in F(O)$ such that $l(o_1) = l(o_2)$, $r(o_1) = r(o_2)$, $cov(o_1) \supseteq cov(o_2)$, and $cost(o_1) < cost(o_2)$, we say that operation o_1 dominates operation o_2. That is, if one operation covers the same set (or a superset) of customers as another operation, but at a lower cost, it is clearly superior and so dominant.

Moreover, for any distinct pair of operations $o_1, o_2 \in F(O)$ such that $l(o_1) = l(o_2)$, $r(o_1) = r(o_2)$, $cov(o_1) = cov(o_2)$, and $cost(o_1) = cost(o_2)$, we arbitrarily declare that the operation with lower index dominates the other. Any operation that is not dominated by another operation is called *dominant*. Therefore, we obtain the set of all dominant operations $D(O), D(O) \subseteq F(O)$, pruning the set of feasible operations $F(O)$.

Splitting of Operations that Service Multiple Customers

The main step of the first phase consists in the generation of a set of operations each one serving exactly one customer. In particular, for any $o \in D(O)$ that services more than one customer, we will split the operation into multiple *clone operations*. That is, for each $o_j \in D(O)$, if $cov(o_j) = \{c_{j,1}, c_{j,2}, ..., c_{j,l}\}$, then $\forall i \in \{1, 2, ..., l\}$ there is an operation $o_{j,i} \in S(O)$ such that $cov(o_{j,i}) = \{c_{j,i}\}$, and where $l(o_{j,i}) = l(o_j), r(o_{j,i}) = r(o_j), cost(o_{j,i}) = cost(o_j)$. The set $S(O)$ is called the set of split operations. In $S(O)$, we effectively store separate copies of an operation for each customer it services. For each pair of clones $o_{j,i_1}, o_{j,i_2} \in S(O)$, we define $lCost_{o_{j,i_1}, o_{j,i_2}} = 0$.

Special Operation for the Depot

If the addressed problem requires the route to start and terminate at a predefined depot location, we add an artificial customer c_0 into the set C corresponding to the depot location. We also construct an artificial operation $o_0 \in S(O)$ where $l(o_0) = c_0$, $r(o_0) = c_0$, $cov(o_0) = \{c_0\}$ and $cost(o_0) = 0$. Moreover, c_0 is not covered by any other operations. As a consequence of this, any feasible solution will necessarily include artificial operation o_0 to ensure that artificial customer c_0 is serviced.

Finally, we highlight that $|S(O)| = \sum_{o \in D(O)} s(o) + 1$, inclusive of the special operation for the depot.

3.4.2 Construct an Equality Generalized TSP

In the second phase, a graph is built on the basis of the set S(O). More precisely, each $o \in S(O)$ will correspond to a vertex. For each $o_1, o_2 \in S(O)$ there is an arc (o_1, o_2) and let x_{o_1, o_2} be a binary variable equal to 1 if o_1 and o_2 are operations that have both been selected as part of our solution, and o_2 is the first operation that occurs after the completion of o_1. Otherwise, $x_{o_1,o_2} = 0$. For each arc (o_1, o_2) , the corresponding cost is $lCost_{o_1,o_2}$. In particular, for any pair of operations $o_1, o_2 \in S(O)$, where o_1 and o_2 are not clones of the same operation, we define:

$$lCost_{o_1,o_2} = cost(o_1) + relCost(r(o_1), l(o_2)),$$

where $relCost(r(o_1), l(o_2))$ is defined as the cost of relocating the truck with all drones on board from the rendezvous point of o_1 to the launch point of o_2. (In problems where we seek to minimize route completion time, this relocation cost is the time required for the truck to drive directly from $r(o_1)$ to $l(o_2)$.) Thus, the cost of linking two consecutive operations is the direct cost of the first operation plus the cost of relocating between consecutive operations.

On this basis, it is possible to recast the truck-and-drone coordination problem as an Equality Generalized Traveling Salesman Problem (E-GTSP). For each customer location $c \in C$, we must visit exactly one vertex o such that $cov(o) = \{c\}$. Moreover, let us define $cluster(c) = \{o \in S(O) : cov(o) = \{c\}\}$. Because we split operations that covered multiple customers, for any pair of customers $c_1, c_2 \in C$, $cluster(c_1) \cap cluster(c_2) = \emptyset$. That is, our clusters are disjoint. We define a cover matrix $covMat$, where the element $covMat_{o,c}$ has value 1 if and only if $cov(o) = \{c\}$ and has value 0 otherwise, $\forall o \in S(O), c \in C$.

We may formulate our problem as follows:

$$\texttt{Minimize:} \sum_{o_1,o_2 \in S(O)} lCost_{o_1,o_2} * x_{o_1,o_2} \tag{3.1}$$

$$\texttt{subject to:} \tag{3.2}$$

$$\sum_{o_1 \in S(O)} x_{o_1,o_2} = \sum_{o_1 \in S(O)} x_{o_2,o_1}, \forall o_2 \in S(O) \tag{3.3}$$

$$\sum_{o \in S(O)} (covMat_{o,c} * (\sum_{o_1 \in S(O)} x_{o_1,o})) = 1, \forall c \in C \tag{3.4}$$

$$x_{o_1,o_2} \in \{0, 1\}, \forall o_1, o_2 \in S(O) \tag{3.5}$$

$$\texttt{subtour elimination} \tag{3.6}$$

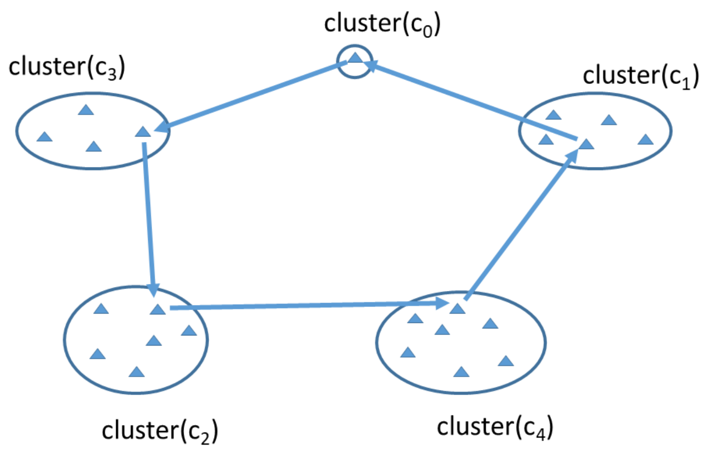


Figure 3.2: A conceptual depiction of a solution of a truck-and-drone coordination problem with four customer locations as an E-GTSP solution.

The objective function (3.1) accounts for all costs during operations and all costs between operations. Constraints (3.3) are ordinary flow constraints. Constraints (3.4) ensure that every customer is serviced (covered) by at least one operation that is visited in the solution path. Constraint (3.5) ensures our decision variables are binary. Constraints (3.6) are generic subtour elimination constraints. A conceptual depiction of a truck-and-drone coordination problem with four customer locations as an E-GTSP is showed in Figure 3.2. Each oval-shaped region defines the boundary of a cluster associated with a particular customer. Each triangle within a cluster represents an operation $o \in S(O)$ that services the customer associated with the cluster. The line segments trace a closed tour that visits at least one triangle (operation) within each cluster. Thus, the tour services all customers. We emphasize that the clusters are disjoint, because no split operation services more than one customer. The depot is represented as customer c_0 and $cluster(c_0)$ consists of one special operation that covers the depot.

3.4.3 Transform the E-GTSP Problem into an Asymmetric TSP

A procedure for transforming any E-GTSP instance into an asymmetric TSP (ATSP) instance with the same number of nodes has previously been described [44]. To apply that procedure in this case, the following steps must be performed:

1. For each $c \in C$, create an arbitrary directed cycle among $cluster(c)$ and set $c'_{o_i,o_j} = 0$ whenever o_j immediately follows o_i in a directed cycle within a cluster.

2. For any $o_i, o_j \in S(O)$ such that $cov(o_i) \neq cov(o_j)$, set $c'_{o_i,o_j} = lCost_{o_k,o_j} + M$, where o_k succeeds o_i in the directed cycle of $cluster(cov(o_i))$. Here, M is a sufficiently large number.

3. Set $c'_{o_i,o_j} = 2M$ for all other edges.

4. Solve the ATSP, where the vertex set is $S(O)$ and the cost between an arbitrary set of vertices $o_i, o_j \in S(O)$ is c'_{o_i,o_j}.

5. Subtract $M|C|$ from the objective value, to get the objective value for the original problem.

Our solution is then extracted by selecting the operation associated with the first node visited within each cluster in the ATSP solution. These selected operations are performed in the same order that the clusters are visited in the ATSP solution. Solving the ATSP can be accomplished using an exact or heuristic approach.

In Figure 3.3, we display an example solution resulting from our algorithm applied a truck-and-drone coordination problem, where $|V| = 50$ and $|C| = 50$. Customer locations are distributed uniformly over a 100km by 100km square region. A feasible operation in this example contains at most one customer delivery by drone. Dashed green and red line segments show the outbound and inbound flight paths of the drone, respectively. Solid black segments shows the path of the truck. Each $v \in V$ or $c \in C$ is shown as a black square. The red circle in the bottom left is the depot location. The drone has a range of 20 minutes. The truck has speed 1 km/minute; the drone has speed 2 km/minute. Both vehicles follow the Euclidean distance metric. Integer labels offset above and to the right of a black square indicate order of delivery. A feasible operation, in this example, allows the drone to carry a

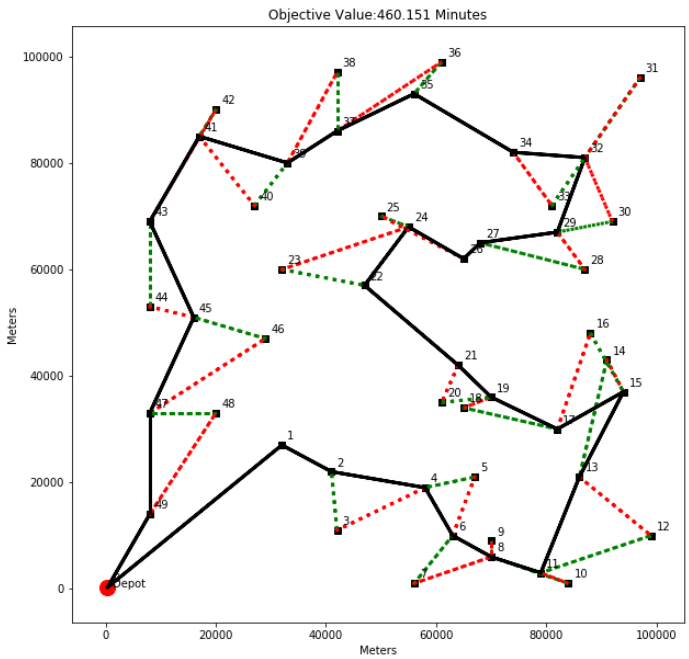


Figure 3.3: An example solution resulting from our algorithm applied a truck-and-drone coordination problem, where $|V| = 50$ and $|C| = 50$.

single package, and the finite flight time of the drone must be respected.

3.5 Analysis of Algorithm

In this section we will analyze some properties of the proposed algorithm. In particular, the optimality of the solution method and the maximum size of the resulting ATSP will be discussed in detail.

3.5.1 Optimality of Solution Method

Theorem 1. *If the ATSP resulting from the problem transformation described in Section 4 is solved exactly, then the algorithm constructs an optimal solution to the truck-and-drone coordination problem under consideration.*

Proof. All feasible operations are initially considered. In any feasible solution, replacing a dominated operation, o_1, by an operation o_2 that dominates o_1 preserves feasibility, as o_1 and o_2 service the same set of customers, and the replacement of o_1 by o_2 never increases the objective value. Thus, an optimal solution exists that is strictly composed of operations contained within $D(O)$.

Next, we note that in the E-GTSP formulation, the objective function is set in a manner that accounts for all costs associated with the tour. Next, we applied a transformation from E-GTSP to ATSP, which has been proven in [44] to produce an equivalent problem.

Therefore, the resulting ATSP is equivalent to the original truck-and-drone coordination problem, and the optimal solution to the ATSP corresponds to an optimal solution of the truck-and-drone coordination problem. □

Remark. *We highlight that if a heuristic solver is used to solve the ATSP resulting from Section 3.4, then the only source of suboptimality is related to the suboptimality of the heuristic solver.*

3.5.2 Bounds on the Size of the Resulting ATSP

Our proposed algorithm requires solving an ATSP instance where the vertex set is $S(O)$. The tractability of the algorithm, therefore, depends heavily on the cardinality of the set $S(O)$. That is, if $S(O)$ is very large, it may be intractable to solve an ATSP over $S(O)$. Thus, we seek to establish bounds on the size of the set $S(O)$ for various problem assumptions.

For each fixed launch point, fixed rendezvous point, and given customer set, there is at most one non-dominated operation for that combination of launch point, rendezvous point, and customer set. Therefore, the number of non-dominated operations can be bounded as follows:

$$|D(O)| \leq |V|^2 * \sum_{i=1}^{MaxS} (|C|!/(|C|-i)!),$$

where $MaxS = max_{o \in F(O)}(s(o))$ is the maximum number of customers that may be visited within any single feasible operation. The above inequality is formed by noting that there are $|V|^2$ pairs of launch/rendezvous points, and for each pair, we could at worst choose any subset of $1 \leq i \leq MaxS$ customers within the entire customer set C. We note that the right-hand side of the above inequality is $O(|V|^2 * |C|^{MaxS})$. By noting that the size of an operation $o \in D(O)$ is represented as i in the above inequality, we can conclude:

$$|S(O)| \leq |V|^2 * \sum_{i=1}^{MaxS} i * (|C|!/(|C| - i)!).$$

In the case that $MaxS = |C|$, it follows that:

$$|D(O)| \leq |V|^2 * (2^{|C|} - 1)$$

and

$$|S(O)| \leq |C| * |V|^2 * (2^{|C|} - 1).$$

In the case that $MaxS = 1$, then:

$$|S(O)| = |D(O)| \leq |V|^2 * |C|.$$

Let us define $reach(l, r)$ as the set of customers that are reachable (i.e., could feasibly be serviced) by an operation that begins at l and terminates at r. We could tighten our bounds as follows:

$$|D(O)| \leq \sum_{l \in V} \sum_{r \in V} \sum_{i=1}^{min(MaxS, reach(l,r))} (|reach(l, r)|!/(|reach(l, r)| - i)!$$

and

$$|S(O)| \leq \sum_{l \in V} \sum_{r \in V} \sum_{i=1}^{min(MaxS, reach(l,r))} i * (|reach(l, r)|!/(|reach(l, r)| - i)!.$$

That is, for any launch point (l) and rendezvous point (r) combination, we can select any subset of customers of size $1 \leq i \leq min(MaxS, reach(l, r))$ from the set $reach(l, r)$ to be serviced in the operation.

3.6 Computational Results

Instance and solution data for our computational results is available upon request from the corresponding author. Visual solutions for each instance solved are also available. All computations were performed on the same computer with an Intel i7-6700 processor operating at 3.4 GHz and containing 16 GB of RAM. Python 2.7 was used to implement code.

3.6.1 Single Truck Instances: Exact Solver vs. LKH2 for ATSP

We tested our proposed algorithm on benchmark instances [7], which are described in [1]. In each instance, $V = C$. Throughout Euclidean distances are used. In Table 3.1, we apply our algorithm to a truck-and-drone coordination problem and compare the use of an exact ATSP solver and heuristic ATSP solver. In this truck-and-drone coordination problem, a drone is allowed to visit a single customer. After launch of the drone, the truck is assumed to drive directly to the rendezvous location. The truck moves at unit speed. The drone moves at twice the speed of the truck and has a maximum battery duration of 20 time units, which must be respected. Each row in Table 3.1 represents averages over ten instances. Time is in seconds.

The column titled Size reports the value of $|V| = |C|$. The column Type describes the distribution of customers in the instances. In the Uniform type, depot and customer locations were randomly selected from a 100 by 100 square region. The distribution for the 1-center instances exhibits radial symmetry around the origin, (0,0), which is meant to simulate a city center. The distance from the origin is normally distributed with mean of 0 and standard deviation of 50. The 2-center instances are generated in the same manner as the 1-center instances, except that there is a 0.5 probability of a displacement of 200 units along the x-axis. In other words, there are two city centers at (0,0) and (200,0).

The columns under Exact Solver refer to measurements observed when we used an exact formulation in Gurobi 8.1.0 to solve the ATSP resulting from the problem transformation. Likewise, the columns under LKH2 Solver refer to measurements observed when LKH2 [34] (which is an implementation of [37]) was used as the solver for the resulting ATSP. All default settings of LKH2 were used, except we used five independent runs. Columns titled Obj and Time state the average objective value and computation time, respectively, for a given ATSP solution method. The column ATSP Size refers to the average cardinality of the set $S(O)$, and thus the average number of nodes contained in the resulting ATSP. The column Gap% measures the suboptimality of the solutions yielded by using LKH2 as the ATSP solver (i.e., (LKH2 Obj - Exact Obj)/Exact Obj). Preliminary testing indicated that the exact solver was incapable of solving instances of size 50 or 75 within a reasonable amount of time (i.e., within five hours). In these cases, we reported a dash (-) in the corresponding table entries for Obj and Time. If an exact objective value was not found for a row instances, then Gap% could not be computed and a dash (-) was placed in the corresponding row in the Gap% column. We highlight that when using the exact solver for the ATSP resulting from the problem transformation, optimal solutions are found. On instance sets where we know the optimal solutions, using LKH2 as the ATSP solver resulted in optimality gaps from 0.00% to 0.07%.

Instance		Exact Solver		LKH2 Solver		ATSP Size	Gap%
Size	Type	Obj	Time	Obj	Time		
10	Uniform	301.12	0.127	301.13	0.309	21.1	0
10	1-center	431.44	0.186	431.45	0.216	30.7	0
10	2-center	714.31	0.092	714.31	0.234	17.9	0
20	Uniform	364.42	9.085	364.67	0.613	113	0.07
20	1-center	581.71	51.118	581.86	1.612	226.1	0.03
20	2-center	790.08	2.27	790.28	0.383	85.7	0.03
50	Uniform	-	-	455.98	81.457	1394.8	-
50	1-center	-	-	757.58	605.892	4169	-
50	2-center	-	-	1203.13	38.364	925.4	-
75	Uniform	-	-	525.54	1339.36	5349.3	-
75	1-center	-	-	1045.68	5448.51	10857.4	-
75	2-center	-	-	1460.38	708.976	4124.1	-

Table 3.1: Objective values, computation time and ATSP size for our algorithm applied to a truck-and-drone coordination problem.

3.6.2 Analysis of Computational Results

In Table 3.1, we note that we were able to find exact solutions within a minute on average to all instances of size 20 or less. However, once the problem size rose to 50, the exact solver became intractable. We also note that among the instances for which we know the optimal solution, the average gap by applying LKH2 as the ATSP solver varied between 0.00% and 0.07%. The maximum gap on any individual instance was 0.6%. LKH2 has

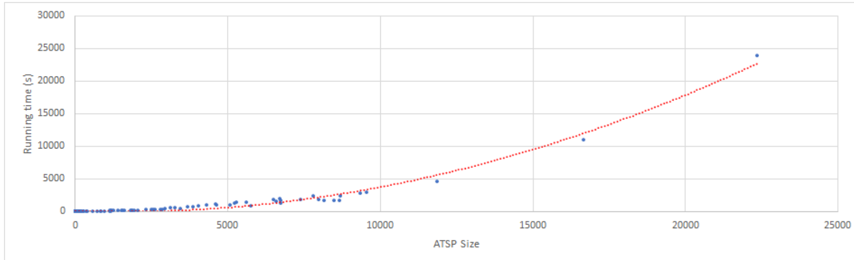


Figure 3.4: Computational times, when using LKH2 as the solver, appears to scale quadratically with the ATSP Size.

been extensively benchmarked across a variety of TSP and ATSP problems, typically yielding solutions very near to optimal. [34] The small optimality gaps observed in Table 3.1 suggest that the proposed transformation does not result in an unusual ATSP structure that causes LKH2 to perform poorly.

We also solved instances of size 50 and 75 using the LKH2 heuristic, but we would like to point one implementation detail especially relevant to larger instances. Explicitly storing the full matrices for $lCost$ and c', which are $O(|S(O)|^2)$, can cause a computational bottleneck due to memory usage. However, it is possible to compute each value of c' without ever explicitly storing $lCost$. Instead, each element of c' can be computed on the fly if we know the cost of the operations ($O(|S(O)|)$), the relocation cost between any pair of vertices in V ($O(|V|^2)$), and cluster memberships of each split operation ($O(|S(O)|)$), which collectively require much less memory ($O(max(|V|^2, |S(O)|)$).

Computational times predictably increased with instance size. For a fixed size instance, the 1-center instances had the largest computational times (with the exception when Size=10). The 1-center instances had the largest number of feasible operations, as the customer locations display the greatest level of clustering. Therefore, the average ATSP Size was also larger. In general ATSP Size was closely associated with total computational time when using LKH2 as the solver. In Figure 3.4, we can see the relationship is nearly quadratic in nature.

The distribution of customer locations was impactful. As an example, in

1-center instances, the customer density in the circle of radius 10 around the origin (0,0) was more than five times larger than the density of customers in the uniform distribution. This resulted in a larger number of feasible operations relative to uniform and 2-center instances. This can be seen by observing that the value in the column ATSP Size is larger for 1-center instances than it is for uniform or 2-center instances. Additionally, we observed that the computational time was typically right-skewed, especially for 1-center instances (e.g., median computation time for 1-center instances of size 75 was 2560 seconds; the mean was 5449 seconds). We relate this to the fact that if the density of launch, retrieval, and customer sites are all multiplied by a factor of n within a small region, we would expect the number of feasible operations to rise by approximately a factor of n^3. Even small variability in the size of the central cluster in a 1-center instance can have an approximately cubic impact on the number of feasible operations, which consequently affects computational time.

Lastly, we point out that we always used five independent runs when calling the LKH2. If computation time is at a premium, we could reduce the number of independent runs. If we wish to increase confidence in the quality of the solutions, we may increase the number of runs.

3.7 Conclusions

We described a framework for transforming a broad swath of truck-and-drone coordination problems first into equality generalized traveling salesman problems and consequently into asymmetric traveling salesman problems. This framework can be extended to cases where multiple trucks are allowed. The resulting ATSP can then be solved by existing solvers. The key requirement is that the set of feasible operations for the truck-and-drone coordination problem can be enumerated.

We applied this transformation to a test truck-and-drone coordination problem. We were able to find optimal solutions to problems with 20 uniformly distributed customers in 9 seconds, on average. By replacing the exact ATSP solver with a heuristic solver [34], we found solutions that were between 0.00% and 0.07% suboptimal, depending on the characteristics of the instance. The heuristic solver significantly decreased computation times and allowed us to solve significantly larger instances. Due to the extensive benchmarking of LKH2 and small gaps on instances where we know the optimal solution, we are encouraged that applying this heuristic solver on transformed problems does not result in significantly suboptimal solutions.

We also found that computation time was strongly correlated with the cardinality of the set of split operations, $S(O)$. We used five independent runs in LKH2, which could be reduced if computation time is excessively costly.

We would like to conduct further testing of our framework applied to several other problem types, including the Flying Sidekick Traveling Salesman Problem [43], the min-time Traveling Salesman Problem with Drone (TSP-D) [1], and min-cost TSP-D [32], which all allow the truck to visit customers while the drone is airborne. We would also like to study problems which allow multiple drones per truck including the k-Multi-visit Drone Routing Problem (k-MVDRP)[47], which allows k drones per truck where the drones can potentially visit multiple customers in a single flight, and the Multiple Flying Sidekicks Traveling Salesman Problem [42]. An extensive computational study comparing objective values across problem types and various geometries of customer locations and street networks might provide valuable insights.

Due to the strong link between $|S(O)|$ and computation time, we would like to explore mechanisms to prune the set of operations considered. The goal would be to effectively reduce the cardinality of $S(O)$ and, therefore, computation time, while maintaining high solution quality. An analysis of operations that appear in the optimal solution versus operations that do not appear in the optimal solution may shed light on the question: "What are the characteristics of a 'good' operation?" Other potential research directions can be the use of a divide-and-conquer strategy to solve smaller subproblems or the integration of the proposed method within a column generation approach. In particular, the combination of the proposed method and a column generation approach could result in a very effective method since the operations would be generated on the fly. In this way, the drawback related to the quadratic growth of the number of operation with the size of the problem could be partially overcome.

More generally, by recasting truck-and-drone coordination problems as traveling salesman problems, we can explore what methodologies for the ordinary traveling salesman problem can now provide value to transformed truck-and-drone coordination problems.

Chapter 4

An exact solution method for the Flying Sidekick Traveling Salesman Problem

4.1 Introduction

In the last years, the attention of the operations research community towards optimization problems with drones has continuously grown, as reported in [45]. The motivations leading to this level of attention were mainly related to the new developments in drone technology and the subsequent exploration by large corporations such as Amazon, Google, UPS, and DHL of using drones [5, 14, 56, 57, 60].

One of the most promising application field where the use of drones can result useful is the last-mile logistic. Indeed, several studies showed the benefits, in terms both of emissions and mainly of completion time reduction, that can be achieved using drones for parcel deliveries. The drone delivery system most studied in literature is the one consisting of a truck and a drone. The truck, other than servicing customers, can carry one or more drones and acts as a mobile depot for them (replenishing their batteries and providing the parcels that have to be delivered). The drone is launched from the truck to serve some customers and then it comes back to the truck to pick up a new parcel to deliver or to the depot. The first problem which considered this hybrid working unit to make the delivery process more efficient and, possibly, less expensive was the Flying Sidekick Traveling Salesman Problem (FSTSP). After that, several variants of this problem were defined but the majority of these studies put scarce attention to the exact solution of the addressed problem. Indeed, their main focus usually consists of the development of a heuristic method able to tackle effectively the problem. In this chapter, a study of the exact solution of the FSTSP will be presented.

In particular, the chapter is organized as follows: in Section 4.2 the studies focusing on the exact solution of delivery problems with drones will be recalled; In Section 4.3 a detailed description of the problem setting is given together with a new Mixed integer Linear Programming Formulation that, contrarily to the formulations in literature, doesn't use $BigM$ constraints. Moreover, some valid inequalities are presented within the same section; in Section 4.4 an original row-and-column generation approach is described; Section 4.5 proves the effectiveness of the proposed approach showing some computational results; finally, conclusions are given and future perspectives are discussed in Section 4.6.

4.2 Related works

A large body of literature has been dedicated to optimization problems with drones; however, in this section, we will report only those studies that are highly related to the FSTSP or those in which the focus is on the exact solution of the tackled problem.

The first definition of the FSTSP is given in [43]. The authors, in addition the problem definition, presented a Mixed Integer Linear Programming formulation and a set of benchmark instance of up to 10 nodes. The proposed formulation wasn't solvable within a time limit of half an hour for some of the proposed instances.

A problem similar to the FSTSP, and named TSPD, is defined in [1]. In this paper, a mathematical programming formulation is proposed together with two heuristic algorithms and a new set of benchmark instances of up to 50 nodes. The formulation was solvable only for some of the instances with 10 nodes. The same authors proposed an exact solution method based on dynamic programming in [8]. The authors showed through an extensive numerical analysis that the proposed approach was more effective than the solution of the integer programming proposed in their previous work.

A slightly different problem, named $min - cost\ TSPD$, was defined in [32]. The main difference is represented by the objective function that minimizes the operational costs arising from the distance traveled by the two vehicles and the waiting times of a vehicle for the other one. An original mixed integer linear programming formulation is also proposed but only instances up to 10 nodes can be solved to optimality.

A study very focused on the optimal solution of the TSPD is reported in [30]. The authors presented a new mathematical formulation for the problem. Moreover, they enhanced the tractability of the proposed MIP formulation through a combination of valid inequalities, pre-processing, and other bound tightening strategies. Computational results, carried on the instances provided in [1], showed that the proposed formulation was able to solve instances up to 24 nodes.

Two original formulations are proposed in [26] and solved through a Branch-and-Cut algorithm. The authors first reformulate the model proposed in [43] and then they substitute some explicit constraints of this model with exponentially many constraints so obtaining the two formulations proposed. Moreover, the authors proposed a new set of valid inequalities to tighten their formulations and showed that the proposed formulations are able to solve instances of up to 12 customers.

4.3 Problem description and formulation

The FSTSP can be seen as a particular variant of the TSP (in Figure 4.2 an example of the solutions of these two problems is reported). Solving a TSP instance requires to decide the order according to which the customer will be served minimizing the total route length. On the other hand, solving an FSTSP instance requires to decide:

- The route performed by the truck;
- The set of clients served by the drone where the drone can only deliver parcels to eligible customers (payload capacity);
- The launch node and the pick-up node for each drone flight satisfying the endurance limit of the drone;

with the aim of minimizing the overall delivery time.

To formulate the problem we make exactly the same assumptions as in [43]. More precisely we have that:

1. Each customer must be served once by either the truck or the drone. If the weight of a parcel exceeds the drone payload capacity the corresponding customer can be served only by the truck.

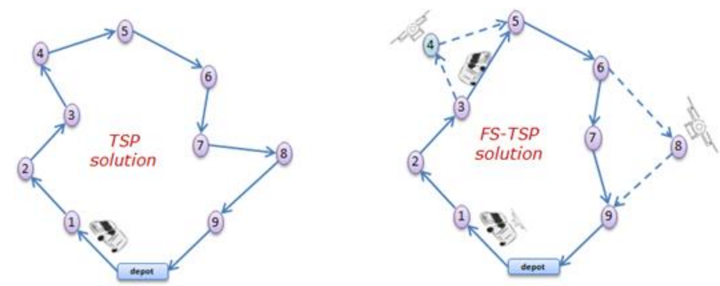


Figure 4.1: A comparison of the TSP and FSTSP solutions

2. The truck and the drone depart and return to a single depot exactly once.
3. The truck has an infinite capacity and acts as a mobile depot for the drone.
4. The drone has a limited endurance in terms of flight time and can serve one customer for flight.
5. Each drone sortie may begin either at the depot or from a customer location.
6. Prior the launch, a service time is required for the driver to change the battery and load the parcel.
7. The second node of a sortie must be a customer served by the drone.
8. The third node may be either the depot or a customer location where the drone meets the truck (each customer cannot be re-visited by the truck just to retrieve the drone).
9. If a sortie ends at a truck location, then another service time is required for the driver to recover the drone.
10. The objective function requires the minimization of the time needed to serve all the customers and come back to the depot.

4.3.1 Extended graph representation

A "natural" representation of an FSTSP instance is given by a multi-graph $G(V; E_T \cup E_D)$ where: $V = \{0, 1, ..., n\}$ is the set of vertices with vertex

0 representing the depot and the other vertices representing the clients; E_T is the set of truck edges; E_D is the set of drone edges. For each truck arc $(i,j)^T \in E_T$ we have a cost $c^T_{(i,j)}$ given by the time needed for the truck to move from the vertex i to the vertex j. For each drone arc $(i,j)^D \in E_D$ we have a cost $c^D_{(i,j)}$ given by the time needed for the drone to move from the vertex i to the vertex j.

In the following, we will use a representation of the FSTSP based on an extended simple-graph $G'(\{s,t\} \cup T \cup D; A_T \cup A_D)$ where:

- the depot $0 \in V$ is splitted into two nodes, an origin node s with only outgoing arcs and a destination node t with only incoming arcs;
- each vertex $i \in V \setminus \{0\}$ is splitted in a truck node $i \in T$ that can be served by the truck and a drone node $i' \in D$ reachable only by the drone;
- for each $i \in \{s\} \cup T$ and each $j \in T \cup \{t\}$ with $i \neq j$, we define a truck arc $(i,j) \in A_T$ whose cost d_{ij} is given by the time needed for the truck to move from node i to node j;
- for each $i \in \{s\} \cup T$ and $j' \in D$, we define a loaded drone arc $(i,j') \in A^L_D$ whose cost $d_{ij'}$ is given by the time needed for the drone to move from node (depot or customer) i to customer j;
- for each $i' \in D$ and $j \in T \cup \{t\}$, we define an empty drone arc $(i',j) \in A^E_D$ whose cost $d_{i'j}$ is given by the time needed for the drone to move from customer i to node (depot or customer) j;
- the set of drone arcs is given by the union of the truck arcs, the loaded drone arcs and the empty drone arcs $A_D = A_T \cup A^L_D \cup A^E_D$.

To better understand the extendend graph, in Figure 4.2 we report the same feasible solution of an FSTSP instance on the original and the corresponding extended graph.

We can observe that any feasible solution on the extended graph consists of a truck path and a drone path from $\{s\}$ to $\{t\}$ and, for each customer i, either node i belongs to the truck path or node i' belongs to the drone path. Moreover, we have that the drone arc (i,j) with $i,j \in T \cup \{s,t\}$ can belong to the drone path if and only if the truck arc (i,j) belongs to the truck path. Similarly, a drone arc (i',j) with $i' \in D$ and $j \in T \cup \{t\}$ can belong to the drone path if and only if node j belongs to the truck path.

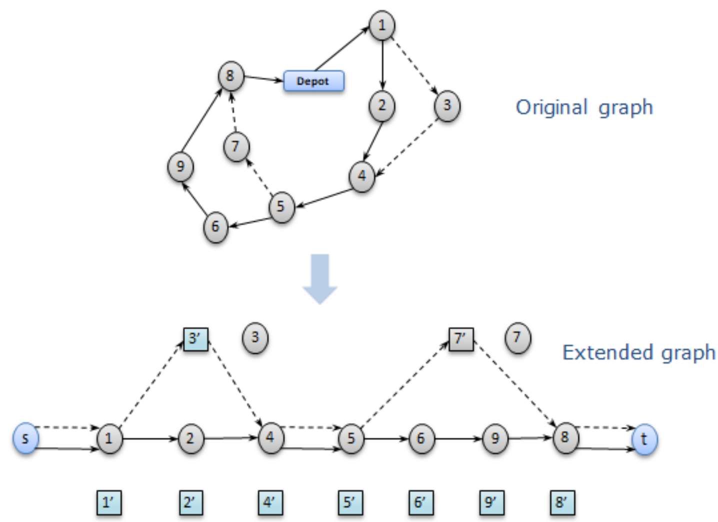


Figure 4.2: An example of feasible solution

On the basis of this observation, it is possible to express the total delivery time of a generic feasible solution as a sum of three terms:

- the duration of the truck route;
- the total launch and delivery time;
- the total truck delivery time.

Let Π_{ij} be the the set of all the paths from i to j on the truck subgraph $G'_T(\{s,t\} \cup T; A_T)$, to evaluate the total delivery time of a feasible solution we can consider the set of couples $S = \{(k', P_{ij})|k' \in D, i, j \in T \cup \{s,t\}\ and\ Pij \in \Pi_{ij}\}$ where each couple (k', P_{ij}) denotes a sortie in which the drone travels the path $i - k' - j$ and the truck travels the path P_{ij}. A sortie (k', P_{ij}) is feasible if either the drone path $i - k' - j$ and the truck path P_{ij} have a duration less than the endurance of the drone Dtl (drone time limit). Moreover, we say that it is a waiting truck sortie if it is feasible and the duration of the drone path is greater than the duration of the truck path. For each waiting truck sortie we denote by $W^k_{P_{ij}}$ the truck waiting time.

4.3.2 Problem formulation

To introduce the new formulation for the FSTSP, we define three sets of binary variables:

- y_{ij}, for each $(i, j) \in A_T$ is equal to 1 if the arc (i, j) belongs to the truck path, 0 otherwise.
- x_{ij}, for each $(i, j) \in A_D$ is equal to 1 if the arc (i, j) belongs to the drone path, 0 otherwise.
- $z^k_{P_{ij}}$, for each waiting time sortie ($(k', P_{ij}) \in S$ such that $W^k_{P_{ij}} > 0$) is equal to 1 if the drone travels the path $i - k' - j$ and the truck travels the path P_{ij}, 0 otherwise.

By using these variables, the **objective function** is:

$$\sum_{(i,j)\in A_T} d_{ij} y_{ij} + \sum_{(i,j')\in A^L_D} (SL + SR) x_{ij'} + \sum_{(k',P_{ij})\in S | W^k_{P_{ij}} > 0} W^k_{P_{ij}} z^k_{P_{ij}}$$

where SL and SR are the launch and the recovery service time, respectively. The first term measures the duration of the truck path, the second term the total service and recovery time and the third term measures the total truck waiting time.

The set of constraints can be divided into seven subsets:

Single assignment constraints

$$\sum_{i:(i,j')\in A^L_D} x_{ij'} + \sum_{i:(i,j)\in A_T} y_{ij} = 1 \qquad j \in T \qquad (4.1)$$

They impose that each client must be served either by the drone or by the truck.

Consistency constraints

$$\sum_{j':(i,j')\in A^L_D} x_{ij'} \leq \sum_{j:(i,j)\in A_T} y_{ij} \qquad i \in T \qquad (4.2)$$

$$x_{ij} \leq y_{ij} \qquad (i, j) \in A_T \qquad (4.3)$$

Constraints (4.2) impose that every drone sortie has to start from a node i belonging to the truck path. Constraints (4.3) enforce that an arc $(i, j) \in A_t$ can belong to the drone path iff (i, j) belongs also to the truck path.

Truck routing constraints

$$\sum_{j:(s,j)\in A_T} y_{sj} = \sum_{i:(i,t)\in A_T} y_{it} = 1 \tag{4.4}$$

$$\sum_{j:(i,j)\in A_T} y_{ij} = \sum_{j:(j,i)\in A_T} y_{ji} \quad i \in T \tag{4.5}$$

$$\sum_{i,j\in S|(i,j)\in A_T} y_{ij} \leq |S| - 1 \quad S \subseteq T \tag{4.6}$$

They impose that in any feasible solution there is a truck path from the origin node s to the destination node t. Constraints (4.6) are the subtour elimination constraints.

Drone routing constraints

$$\sum_{j:(s,j)\in A_D} x_{sj} = \sum_{i:(i,t)\in A_D} x_{it} = 1 \tag{4.7}$$

$$\sum_{j:(i,j)\in A_D} x_{ij} = \sum_{j:(j,i)\in A_D} x_{ji} \quad i \in T \tag{4.8}$$

$$\sum_{i,j\in S|(i,j)\in A_T} x_{ij} \leq |S| - 1 \quad S \subseteq T \cup D \tag{4.9}$$

They impose that in any feasible solution there is a drone path from the origin node s to the destination node t. Similarly to constraints (4.6), constraints (4.9) represent subtour elimination constraint for the drone path. We highlight that these constraints are not necessary, but they can be used to tighten the formulation.

Backward constraints

$$\sum_{(u,v)\in P_{ij}} y_{ij} + x_{jk'} + x_{k'i} \leq |P_{ij}| + 1 \quad k' \in D, i, j \in T \cup \{s,t\}, Pij \in \Pi_{ij} \tag{4.10}$$

where $|P_{ij}|$ denotes the number of arcs in the path P_{ij}.
The backward constraints (4.10) are necessary to forbid backward sorties like the one highlighted in Figure 4.3.

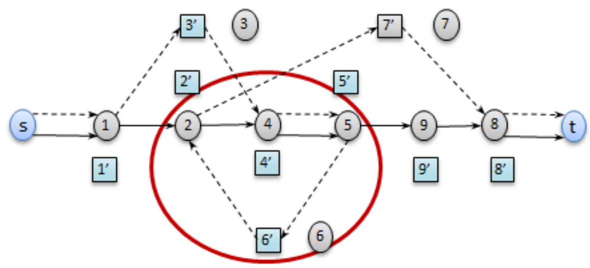


Figure 4.3: An example of backward sortie

Drone endurance constraints

$$x_{ik'} + x_{k'j} \leq 1 \quad i \in \{s\} \cup T, j \in T \cup \{t\}, k' \in D : d_{ik'} + d_{k'j} > Dtl - SR \tag{4.11}$$

$$\sum_{(u,v) \in P_{ij}} y_{uv} + x_{ik'} + x_{k'j} \leq |P_{ij}| + 1 \qquad (k', P_{ij}) \in S : D(P_{ij}) > Dtl - SR \tag{4.12}$$

where $D(P_{ij}) = \sum_{(u,v) \in P_{ij}} d_{uv}$ denotes the duration of the truck path P_{ij}. The drone endurance constraints (4.11, 4.12) impose that the duration of each sortie must be less tha or equal to the drone endurance.

Waiting time constraints

$$z^k_{P_{ij}} \geq \sum_{(u,v) \in P_{ij}} y_{uv} + x_{ik'} + x_{k'j} - |P_{ij}| - 1 \qquad (k', P_{ij}) \in S : W^k_{P_{ij}} > 0 \tag{4.13}$$

Waiting time constraints (4.13) impose that if the drone makes the sortie (k', P_{ij}) then the corresponding variable $z^k_{P_{ij}}$ must be equal to 1 to take into account the truck waiting time in the objective function. Obviously, $z^k_{P_{ij}}$ can be equal to zero, if the corresponding sortie does not provide a truck waiting time ($W^k_{P_{ij}} = 0$).

4.3.3 Preprocessing

The performance of the formulation solution can be improved applying some simple preprocessings that exploit the drone endurance. In particular, let

$(i, k') \in A_D^L$ be a loaded drone arc and let $d_{ik'}$ be its drone travel time. If $d_{ik'} + d_{k'j} > Dtl - SR$ for each node $j \in T \cup \{t\}$ then $x_{ik'}$ is equal to 0 in any feasible solution.

Similarly, if we consider an empty drone arc $(k', j) \in A_D^E$ such that $d_{k'j} + d_{ik'} > Dtl - SR$ for each node $i \in \{s\} \cup T$, we can conclude that $x_{k'j} = 0$ in any feasible solution.

Moreover, let us consider a travel arc $(i, j) \in A_T$ whose truck travel time d_{ij} is greater than $Dtl - SR$. Due to the triangle inequality no one sortie (k', P_{ij}) can be in a feasible solution and then the inequalities $x_{ik'} + x_{k'j} <= 1$ for each $k' \in D$ are valid for the problem.

4.3.4 Valid inequalities

To strengthen the formulation we added to it, a set of valid inequalities.

Proposition 1. *Let $(i, k') \in A_D^L$ be a loaded drone arch and let $T_{ik'} \subseteq T \cup \{t\}$ be the subset of nodes for which a feasible sortie (k', P_{ij}) exists. Due to the triangle inequality, $T_{ik'} = \{j \in T \cup \{t\} : d_{ik'} + d_{k'j} <= Dtl - SR\}$. It is trivial to prove that the inequality:*

$$x_{ik'} \leq \sum_{j \in T_{ik'}} x_{k'j} \tag{4.14}$$

is valid for the FSTSP problem.

Similarly, let us consider an empty drone arch $(k', j) \in A_D^E$ and the subset of nodes $T_{k'j} = \{i \in \{s\} \cup T : d_{ik'} + d_{k'j} <= Dtl - SR\}$. The inequality:

$$x_{k'j} \leq \sum_{i \in T_{k'j}} x_{ik'} \tag{4.15}$$

is valid for the FSTSP problem.

Proposition 2. *Consider three nodes i, σ, j such that $i \in \{s\} \cup T$, $\sigma \in T$ and $j \in T \cup \{t\}$. If $d_{i\sigma} + d_{\sigma j} > Dtl - SR$, no sortie (k', P_{ij}) can be in a feasible solution, if (i, σ) or (σ, j) belongs to the path P_{ij}. Therefore, the inequalities:*

$$x_{ik'} + x_{k'j} + \sum_{\sigma \in S_{ij}} y_{i\sigma} \leq 2 \qquad k' \in D \tag{4.16}$$

$$x_{ik'} + x_{k'j} + \sum_{\sigma \in S_{ij}} y_{\sigma j} \leq 2 \qquad k' \in D \tag{4.17}$$

are valid for the FSTSP problem where $S_{ij} = \{\sigma \in T : d_{i\sigma} + d_{\sigma j} > Dtl - SR\}$.

Moreover, let $P_{ij} = \{i, s_1, ..., s_n, j\}$ be a truck path from node i to node j where $S = \{s_1, ..., s_n\}$ is the set of intermediate nodes.

Proposition 3. *If* $D(P_{ij}) > Dtl - SR$*, then at least an arc or a node of the path must be touched by the drone. Hence, the inequalities:*

$$\sum_{(u,v) \in P_{ij}} y_{uv} \leq |P_{ij}| - 1 + x_{is_1} + \sum_{s \in S} \sum_{k' \in D} x_{k's} \tag{4.18}$$

are valid for the FSTSP problem.

For $|P_{ij}| = 1$ inequality (4.18) begins

$$y_{ij} \leq x_{ij}$$

.

For $|P_{ij}| = 2$ inequality (4.18) begins

$$y_{is} + y_{sj} \leq 1 + x_{is} + \sum_{k' \in D} x_{k's}$$

.

Proposition 4. *The inequalities:*

$$\sum_{(u,v) \in P_{ij}} y_{uv} + x_{ik'} + x_{q'j} \leq |P_{ij}| + 1 + \sum_{s \in S} x_{k's} \qquad k', q' \in D : k' \neq q' \tag{4.19}$$

are valid for the FSTSP problem.

For $|P_{ij}| = 1$ inequality (4.19) begins

$$y_{ij} + x_{ik'} + x_{q'j} \leq 2 \tag{4.20}$$

For $|P_{ij}| = 2$ inequality (4.18) begins

$$y_{is} + y_{sj} + x_{ik'} + x_{q'j} \leq 3 + x_{k's} \tag{4.21}$$

Inequalities (4.20) can be lifted obtaining the following valid inequalities:

$$y_{ij} + x_{ik'} + \sum_{q' \in D : q' != k'} x_{q'j} \leq 2 \tag{4.22}$$

$$y_{ij} + \sum_{q' \in D : q' != k'} x_{iq'} + x_{k'j} \leq 2 \tag{4.23}$$

If $d_{ik'} + d_{k'j} > Dtl - SR$, we can further strengthen inequalities (4.22) and (4.23) obtaining the valid inequalities:

$$y_{ij} + x_{ik'} + \sum_{q' \in D} x_{q'j} \leq 2 \tag{4.24}$$

$$y_{ij} + \sum_{q' \in D} x_{iq'} + x_{k'j} \leq 2 \tag{4.25}$$

Inequalities (4.21) can be lifted obtaining the following valid inequality:

$$y_{is} + y_{sj} + x_{ik'} + \sum_{q' \in D : q' != k'} x_{q'j} \leq 3 + x_{k's} \tag{4.26}$$

If $d_{ik'} + d_{k'j} > Dtl - SR$, then we can further strengthen inequalities (4.26) obtaining:

$$y_{is} + y_{sj} + x_{ik'} + \sum_{q' \in D : q' != k'} x_{q'j} \leq 3 \tag{4.27}$$

4.3.5 Symmetry breaking constraint

In our computational experience, we worked on a symmetric graph so in the enumeration tree we had many symmetries. To reduce these symmetries, we can add the inequality:

$$\sum_{j:(s,j) \in A_T} d_{sj} y_{sj} \leq \sum_{i:(i,t) \in A_T} d_{it} y_{it} \tag{4.28}$$

The problem is that adding these constraints, we could not consider the launch time SL if the sortie starts from the depot. Therefore, we have to add a new variable defined by the following constraints:

$$\theta \leq \sum_{k':(k',t)\in A_D} x_{k't} \tag{4.29}$$

$$1-\theta \geq \sum_{k':(s,k')\in A_D} x_{sk'} \tag{4.30}$$

The variable has coefficient equal to $-SL$ in the objective function.

4.3.6 A variant of the FSTSP

The definition of the FSTSP does not include the possibility for the truck of waiting for drone retrieval in the same location where the drone is launched. Instead, this capability is provided by the Min-cost TSPD where the objective function minimizes the operational costs ([32]). In this subsection, we address a new problem called FSTSP* that is a generalization of the FSTSP in which the drone can be launched and recovered in the same node but, contrarily to the Min-cost TSPD, the objective is still the completion time minimization. To formulate this problem on the basis of the formulation previously described, we have to introduce a new set of binary variable. These variables are the assignment variable σ_{ij} for each $(i,j) \in A_D$, where $\sigma_{ij} = 1$, if the client j is served by the drone starting from and returning to the same node. The coefficient of a generic variable σ_{ij} in the objective function is $c^{\sigma}_{ij} = s_l + s_r + d_{ij} + d_{ji}$. Obviously, if $s_r + d_{ij} + d_{ji} > Dtl$, then the client j cannot be assigned to the node i and $\sigma_{ij} = 0$. Therefore, the new objective function is the following:

$$\sum_{(i,j)\in A_T} d_{ij}y_{ij} + \sum_{(i,j')\in A_D^L} (SL+SR)x_{ij'} +$$
$$+ \sum_{(k',P_{ij})\in S|W^k_{P_{ij}}>0} W^k_{P_{ij}} z^k_{P_{ij}} + \sum_{(i,j')\in A_D^L} c^{\sigma}_{ij'}\sigma_{ij'}$$

Moreover, in addition to the objective function, we also need to modify the single assignment constraints (4.31) as follows:

$$\sum_{i:(i,j')\in A_D^L} x_{ij'} + \sum_{i:(i,j')\in A_D^L} \sigma_{ij'} + \sum_{i:(i,j)\in A_T} y_{ij} = 1 \qquad j \in T \tag{4.31}$$

These new constraints impose that each client must be served either by the truck, by the drone with a standard sortie or by the drone with a cyclic sortie.

4.4 A row-and-column generation approach

To solve to optimality the FSTSP we developed a row-and-column generation procedure and we embedded it into a Branch-and-Cut framework. The column generation procedure does not require to solve a pricing problem so it would be wrong to consider it a Branch-and-Cut-and-Price algorithm. Therefore, a Branch-and-Cut algorithm is adopted for the solution of the proposed model strengthened through the addition of the valid inequalities (4.14, 4.15, 4.16 and 4.17). More precisely, at the root node of the enumeration tree, we solve the LP relaxation where:

i) the preprocessing constraints are added to the formulation;

ii) the valid inequalities (4.14, 4.15, 4.16 and 4.17) are added to the formulation;

iii) the symmetry breaking constraint and the variable θ are added to the formulation;

iv) the set of the new valid inequalities is added to the formulation;

v) the set of $z-$variables $(z^k_{P_{ij}})$ with $|P_{ij}| \geq 3$ are excluded from the LP relaxation;

vi) the subtour elimination constraints (4.6) and (4.9) are relaxed;

vii) the backward constraints (4.10), the drone endurance constraints (4.12) and the waiting time constraints (4.13) with $|P_{ij}| \geq 3$ are relaxed.

Then, at each node of the enumeration tree:

i) we separate the subtour elimination constraints by using a max flow separation procedure;

ii) if an integer solution is found, we separate the endurance and the backtrack constraints;

iii) if a feasible integer solution is found for each sortie $(i - k - j)$ such that $W^k_{P_{ij}} > 0$, we add the variable $z^k_{P_{ij}}$ with related cost $W^k_{P_{ij}}$ and the corresponding waiting time constraint.

4.5 Computational results

In this section, the results of the numerical experiments, aimed at evaluating the effectiveness of the new set of valid inequalities and the performance of the branch-and-cut algorithm, are presented. The experiments have been performed on an Intel(R) Core(TM) i7-8700k, 3.70 GHz, 16.00 GB of RAM. The branch-and-cut algorithm has been coded in C language using Cplex 12.7 with default setting as MIP solver, with a time limit of 1 hour. The experiments have been carried out on two test beds:

- the set of benchmark instances proposed in [43] (Murray's instances);
- a set of 30 new problem instances, named from $unina101$ to $unina2010$ (Unina's instances).

For the sake of completeness, we recall the main features of the set of benchmark instances. The set of customers is distributed across an 8-mile square region and its cardinality is equal to 10. The depot location is randomly chosen to be either the average of the x- and y-coordinates of the customers (i.e., near the center of gravity), the average of the customers' x-coordinates with a y-coordinate of zero, or at the southwest corner of the region (origin). Between 80$\%$ and 90$\%$ of the customers were designated as being UAV-eligible. The drone time limit was chosen to be either 20 or 40 min. UAV speeds were selected as 15, 25, or 35 miles/h, with drone travel being based on Euclidean metric. The truck speed was set to be equal to 25 miles/h, with Manhattan truck travel paths. The overhead parameters SL and SR were assumed to be one minute each.

Moreover, the 30 new instances were generated defining 10 problem instances for three values of the customer set cardinality ($C = \{10, 15, 20\}$). The customer and depot locations have been determined coherently with the instances proposed in [43]. The percentage of customer UAV-eligible is between 80$\%$ and 90$\%$. The drone time limit was selected as 13.34, 20 or 40 min. Truck and drone speeds was chosen to be either 20 or 40 miles/h. The parameters SL and SR were assumed to be one minute each.

4.5.1 Murray's Instances

The computational experiments carried on the benchmark instances were aimed to evaluate the efficacy of the new set of valid inequalities and the breaking symmetry constraint, in addition to the assessing of the proposed solution method. For this reason, we compared two version of the proposed method. The first version called $B\&P1.0$ is the algorithm described in Section 4.4 without the addition of the new set of valid inequalities and the breaking symmetry constraint. The second version is exactly the algorithm described in the previous section. The results of the two versions are given in Table 4.1, where the first three columns represent the instance detail (name, number of customer and drone time limit). The fourth column reports the objective function value of the optimal solution. The successive columns show the number of columns added and the running times for $B\&P1.0$ and $B\&P1.1$, respectively. Finally, we reported in the last column of the table the running times of [26] to better evaluate the performance of the proposed method.

In terms of running times, it is possible to observe that on average the $B\&P1.1$ is the best method among the three reported. Moreover, we can observe that both $B\&P1.0$ and $B\&P1.1$, are able to solve all the instances within a time limit of 1 hour. Finally, we note that the number of columns generated by $B\&P1.1$ is significantly lower than the number of columns generate by $B\&P1.0$. These results confirm the effectiveness of the new valid set of inequalities and of the symmetry breaking constraint.

4.5.2 Unina's Instances

The motivation leading us to generate a new set of instances was related to the need of testing the FSTSP*. Indeed, the solutions obtained solving the FSTSP* formulation on Murray's instances didn't present any cyclic sortie. Therefore, we generated these new instances with the aim of finding a FSTSP* solution with a cyclic sortie. We report the results of the Branch-and-Cut specialized for the FSTSP* on the Unina's instances with 10, 15 and 20 customers in Tables 4.2, 4.3 and 4.4, respectively. The first five columns of these tables report the instance details (name, number of customers, truck speed, drone speed and drone time limit). The successive four columns show some information about the solution method (number of columns added, Upper bound, Lower bound and running times). Finally, the last two columns indicate the number of drone sorties and of cyclic sorties in the obtained solution.

On the basis of the reported results, it is possible to observe that, as expected, the number of drone sorties increase when the drone speed is higher than the truck speed. Moreover, a critical parameter is the drone time limit. Indeed, given a combination of truck and drone speeds, reducing the drone

				B&P 1.0		B&P 1.1		Dell'Amico
Name	**#C**	**Dtl**	**OPT**	**#Cols**	**Time**	**#Cols**	**Time**	**Time**
M37V1	10	20	57.45	77	10.32	26	8.51	2.85
M37V2	10	20	53.79	5	1.96	13	4.59	1.247
M37V3	10	20	54.66	17	2.78	6	4.38	12.637
M37V4	10	20	67.46	3	1.03	4	4.30	14.912
M37V5	10	20	51.78	1279	1190.00	228	321.72	3600
M37V6	10	20	48.60	1170	536.11	183	210.34	1905.68
M37V7	10	20	49.58	135	32.77	63	37.66	204.339
M37V8	10	20	62.38	104	36.39	39	47.50	208.581
M37V9	10	20	43.48	1164	1583.12	248	384.38	3600
M37V10	10	20	41.91	924	1001.25	101	168.37	2087.216
M37V11	10	20	42.90	673	42.57	10	26.28	46.48
M37V12	10	20	56.85	203	37.73	58	23.26	74.366
M40V1	10	20	49.43	151	44.10	32	35.24	107.425
M40V2	10	20	51.71	212	60.48	45	39.87	25.198
M40V3	10	20	57.10	162	34.53	32	29.91	13.005
M40V4	10	20	69.90	74	12.59	6	13.53	1.981
M40V5	10	20	45.46	669	525.96	116	169.44	1505.067
M40V6	10	20	44.51	221	60.38	43	33.07	72.531
M40V7	10	20	49.90	107	26.49	25	10.15	39.141
M40V8	10	20	62.70	77	10.73	24	10.97	25.591
M40V9	10	20	42.53	231	76.54	18	21.20	150.344
M40V10	10	20	43.08	183	53.96	6	3.78	20.322
M40V11	10	20	49.20	15	1.94	2	1.26	1.034
M40V12	10	20	62	2	0.65	1	1.43	1.076
M43V1	10	20	69.59	0	0.37	0	0.55	0.577
M43V2	10	20	72.15	0	0.39	0	0.54	0.502
M43V3	10	20	77.34	0	0.17	0	0.34	0.938
M43V4	10	20	90.14	0	0.21	0	0.37	0.433
M43V5	10	20	58.71	804	1226.47	290	307.92	1788.517
M43V6	10	20	59.09	522	347.32	176	96.24	694.346
M43V7	10	20	65.52	122	26.53	16	23.97	22.341
M43V8	10	20	84.81	157	28.68	80	43.78	276.468
M43V9	10	20	46.93	1382	1855.16	409	381.47	3600
M43V10	10	20	47.93	502	132.07	151	49.09	421.915
M43V11	10	20	57.38	55	8.10	31	9.61	5.009
M43V12	10	20	69.20	18	2.77	6	4.82	2.384

Table 4.1: Results of the proposed solution method

					Branch and Cut				Solution	
Name	#C	T. Speed	D. Speed	Dtl	#AddCols	UB	LB	Time	# Flights	# Cyclic fl.
unina101	10	40 km/h	40 Km/h	20 min.	99	52.49	52.49	86.73	3	0
	10	20 km/h	40 Km/h	20 min	17	91.23	91.23	63.15	3	0
	10	20 km/h	40 Km/h	13.34 min	0	102.80	102.80	3.92	3	1
unina102	10	40 km/h	40 Km/h	20 min.	78	54.77	54.77	178.63	2	0
	10	20 km/h	40 Km/h	20 min	21	89.19	89.19	15.39	3	0
	10	20 km/h	40 Km/h	13.34 min	1	104.78	104.78	3.24	2	1
unina103	10	40 km/h	40 Km/h	20 min.	4	53.37	53.37	2.16	2	0
	10	20 km/h	40 Km/h	20 min	0	102.36	102.36	2.95	3	0
	10	20 km/h	40 Km/h	13.34 min	3	102.36	102.36	3.73	3	0
unina104	10	40 km/h	40 Km/h	20 min.	137	54.97	54.97	73.11	2	0
	10	20 km/h	40 Km/h	20 min	12	97.45	97.45	41.20	4	0
	10	20 km/h	40 Km/h	13.34 min	5	103.43	103.43	14.11	3	0
unina105	10	40 km/h	40 Km/h	20 min.	50	55.43	55.43	13.12	2	0
	10	20 km/h	40 Km/h	20 min	17	103.12	103.12	20.02	4	0
	10	20 km/h	40 Km/h	13.34 min	4	104.16	104.16	8.25	4	0
unina106	10	40 km/h	40 Km/h	20 min.	183	60.83	60.83	246.39	1	0
	10	20 km/h	40 Km/h	20 min	17	113.58	113.58	11.11	2	0
	10	20 km/h	40 Km/h	13.34 min	2	122.82	122.82	4.89	2	0
unina107	10	40 km/h	40 Km/h	20 min.	93	58.67	58.67	118.19	2	0
	10	20 km/h	40 Km/h	20 min	0	99.80	99.80	3.73	3	0
	10	20 km/h	40 Km/h	13.34 min	7	111.36	111.36	9.65	3	1
unina108	10	40 km/h	40 Km/h	20 min.	75	58.65	58.65	33.51	1	0
	10	20 km/h	40 Km/h	20 min	10	104.06	104.06	19.93	4	0
	10	20 km/h	40 Km/h	13.34 min	4	114.26	114.26	16.45	4	1
unina109	10	40 km/h	40 Km/h	20 min.	9	57.43	57.43	13.08	2	0
	10	20 km/h	40 Km/h	20 min	10	109.07	109.07	5.93	3	0
	10	20 km/h	40 Km/h	13.34 min	8	109.07	109.07	4.94	3	0
unina1010	10	40 km/h	40 Km/h	20 min.	46	60.40	60.40	94.15	1	0
	10	20 km/h	40 Km/h	20 min	39	108.73	108.73	207.73	4	1
	10	20 km/h	40 Km/h	13.34 min	9	127.02	127.02	15.24	3	1

Table 4.2: FSTSP* results for Unina's instances with 10 customers

					Branch and Cut				Solution	
Name	**#C**	**T. Speed**	**D. Speed**	**Dtl**	**#AddCols**	**UB**	**LB**	**Time**	**# Flights**	**# Cyclic fl.**
unina151	15	40 km/h	40 Km/h	20 min.	101	95.51	95.51	1952.57	3	0
	15	20 km/h	40 Km/h	20 min	10	179.59	179.59	56.10	3	0
	15	20 km/h	40 Km/h	13.34 min	4	191.00	191.00	23.09	4	0
unina152	15	40 km/h	40 Km/h	20 min.	48	96.78	96.78	471.68	4	0
	15	20 km/h	40 Km/h	20 min	13	175.73	175.73	52.04	6	0
	15	20 km/h	40 Km/h	13.34 min	3	7.89	7.89	7.89	5	0
unina153	15	40 km/h	40 Km/h	20 min.	121	99.13	99.13	963.65	4	0
	15	20 km/h	40 Km/h	20 min	13	167.94	167.94	270.59	5	0
	15	20 km/h	40 Km/h	13.34 min	4	189.19	189.19	47.49	4	0
unina154	15	40 km/h	40 Km/h	20 min.	102	93.93	93.93	2509.74	3	0
	15	20 km/h	40 Km/h	20 min	5	161.53	161.53	76.59	8	0
	15	20 km/h	40 Km/h	13.34 min	16	180.43	180.43	64.22	5	1
unina155	15	40 km/h	40 Km/h	20 min.	62	99.80	99.80	369.95	3	0
	15	20 km/h	40 Km/h	20 min	19	184.99	184.99	117.89	5	0
	15	20 km/h	40 Km/h	13.34 min	6	196.77	196.77	28.33	4	0
unina156	15	40 km/h	40 Km/h	20 min.	106	86.52	86.52	91.56	4	0
	15	20 km/h	40 Km/h	20 min	10	149.52	149.52	21.88	5	0
	15	20 km/h	40 Km/h	13.34 min	13	163.11	163.11	13.48	4	1
unina157	15	40 km/h	40 Km/h	20 min.	71	95.81	95.81	245.32	2	0
	15	20 km/h	40 Km/h	20 min	13	180.62	180.62	22.25	3	0
	15	20 km/h	40 Km/h	13.34 min	8	192.23	192.23	31.93	3	0
unina158	15	40 km/h	40 Km/h	20 min.	152	103.20	103.20	1096.63	3	0
	15	20 km/h	40 Km/h	20 min	12	187.35	187.35	27.24	5	0
	15	20 km/h	40 Km/h	13.34 min	4	202.83	202.83	11.93	3	0
unina159	15	40 km/h	40 Km/h	20 min.	161	103.13	103.13	638.41	3	0
	15	20 km/h	40 Km/h	20 min	10	182.95	182.95	15.83	5	0
	15	20 km/h	40 Km/h	13.34 min	4	195.21	195.21	13.48	4	1
unina1510	15	40 km/h	40 Km/h	20 min.	102	108.80	108.80	409.63	3	0
	15	20 km/h	40 Km/h	20 min	17	196.31	196.31	130.29	6	0
	15	20 km/h	40 Km/h	13.34 min	1	202.72	202.72	24.08	5	0

Table 4.3: FSTSP* results for Unina's instances with 15 customers

					Branch and Cut				Solution	
Name	**#C**	**T. Speed**	**D. Speed**	**Dtl**	**#AddCols**	**UB**	**LB**	**Time**	**# Flights**	**# Cyclic fl.**
unina201	20	40 km/h	40 Km/h	20 min.	113	141.90	141.90	3240.61	3	0
	20	20 km/h	40 Km/h	20 min	41	270.51	270.51	2748.25	8	1
	20	20 km/h	40 Km/h	13.34 min	35	270.84	270.84	1233.72	7	0
unina202	20	40 km/h	40 Km/h	20 min.	115	147.92	147.92	6835.47	3	0
	20	20 km/h	40 Km/h	20 min	28	281.26	281.26	6059.70	8	0
	20	20 km/h	40 Km/h	13.34 min	17	293.45	293.45	310.21	5	0
unina203	20	40 km/h	40 Km/h	20 min.	64	144.55	144.55	3864.15	5	0
	20	20 km/h	40 Km/h	20 min	12	253.57	253.57	1704.86	8	0
	20	20 km/h	40 Km/h	13.34 min	2	284.05	284.05	166.67	6	1
unina204	20	40 km/h	40 Km/h	20 min.	84	143.76	143.76	1284.22	5	0
	20	20 km/h	40 Km/h	20 min	13	251.05	251.05	546.15	8	0
	20	20 km/h	40 Km/h	13.34 min	1	287.41	287.41	147.38	4	0
unina205	20	40 km/h	40 Km/h	20 min.	82	144.29	144.29	2060.24	3	0
	20	20 km/h	40 Km/h	20 min	21	257.25	257.25	757.44	6	0
	20	20 km/h	40 Km/h	13.34 min	7	288.11	288.11	195.45	5	1
unina206	20	40 km/h	40 Km/h	20 min.	52	146.76	146.76	578.64	1	0
	20	20 km/h	40 Km/h	20 min	29	265.57	265.57	129.99	3	0
	20	20 km/h	40 Km/h	13.34 min	1	296.70	296.70	57.44	3	1
unina207	20	40 km/h	40 Km/h	20 min.	140	150.50	150.50	4734.04	4	0
	20	20 km/h	40 Km/h	20 min	30	254.54	254.54	3543.75	6	0
	20	20 km/h	40 Km/h	13.34 min	8	279.30	279.30	881.57	3	1
unina208	20	40 km/h	40 Km/h	20 min.	74	142.23	142.23	5849.20	2	0
	20	20 km/h	40 Km/h	20 min	26	267.73	267.73	1941.07	5	0
	20	20 km/h	40 Km/h	13.34 min	0	287.76	287.76	163.62	4	0
unina209	20	40 km/h	40 Km/h	20 min.	99	147.56	147.56	5546.76	4	0
	20	20 km/h	40 Km/h	20 min	10	265.77	265.77	4912.53	9	0
	20	20 km/h	40 Km/h	13.34 min	1	293.74	293.74	562.18	3	0
unina2010	20	40 km/h	40 Km/h	20 min.	110	142.47	142.47	910.59	4	0
	20	20 km/h	40 Km/h	20 min	15	269.93	269.93	627.27	7	0
	20	20 km/h	40 Km/h	13.34 min	5	278.81	278.81	126.03	5	0

Table 4.4: FSTSP* results for Unina's instances with 20 customers

endurance the number of drone sorties decreases. Finally, we can observe that the cyclic sortie is scarcely performed. These result can be explained considering that the main advantage of adopting an hybrid truck and drone delivery system consists in the parallelization of the workload of the two vehicles. This advantage cannot be fully exploited since the truck is waiting stationary for the drone during a cyclic sortie. Therefore, a cyclic sortie arise only if a customer is far from the other customers and once this customer is served by a drone the drone cannot reach no other customer than the one from where it was launched.

4.6 Conclusions

In this study, we proposed a new Mixed Integer Linear Programming formulation for the FSTSP, which unlike previous formulations for this problem in the literature, does not present $BigM$ constraints. Moreover, we introduced a variant of the FSTSP, called FSTSP*, where cyclic sorties are allowed. We developed a Branch-and-Cut algorithm that is able to solve to optimality instances up to 20 customers within a time limit of 1 hour. Moreover, we introduced a new set of valid inequalities and a symmetry breaking constraints. Computational results showed the effectiveness of the proposed contributions.

Future research directions naturally include further refinement of the algorithm. In addition, it may be interesting to extend the proposed formulation to the multi-truck and/or multi-drone version of the same problem.

Chapter 5

The Multi-visit Drone Routing Problem with Edge Launches

5.1 Introduction

A growing body of literature within operations research has considered the use of drones in civilian contexts. In particular, the reported exploration by large corporations such as Amazon, Google, UPS, and DHL of using drones in a commercial delivery context has spurred significant interest in academia [5, 14, 56, 57, 60].

Although some papers, including [16, 62], consider warehouse-direct-to-customer models of drone delivery, many other papers have explored truck-and-drone hybrid schemes of delivery. Many of these papers note that hybrid truck-and-drone models of delivery offer significant potential for cost savings, reduced delivery time, and reduced emissions. This occurs as the larger capacity and range of the truck is complemented by the speed of the drone, and because there exists an ability to parallelize the work that must be done.

Despite the myriad of papers that have explored various models and computational methods related to truck-and-drone routing problems, as far as we are aware, no paper has considered the ability of a drone to launch from or come back to a truck along an edge through a continuous approach. This ability can enlarge the solution space resulting in greater gains compared to those of the traditional truck-and-drone delivery schemes. An example of the benefit of launching along an edge is reported in Figure 5.1. Figure 5.1(a) and Figure 5.1(b) show a solution with node launches and node and edge launches, respectively. Solid lines indicate the path of the truck and dashed lines indicate the flight path of the drone. Figure 5.1(c) shows the maximum flight range of a drone serving customer $C2$ represented by the diameter of the green circle. Therefore, customer $C2$ cannot be served by a drone in a solution with node launches. Figure 5.1(d) shows a feasible solution with edge launches in which customer $C2$ is served by a drone. However, the practicality of this idea depends on the regulatory climate of the country and region in which the delivery model is applied. In any case, it is possible to suppose that on rural and neighborhood roads (or any other roads with low traffic volume) the truck can slow down or even stop to allow a launch or a rendezvous of the drone along that road. Generally, real contexts contain roads with both high and low volumes of traffic, and the proposed contribution can be easily adapted to this scenario, as we will further discuss. In particular, the study reported in this chapter will consider a modified version of the Multi-visit Drone Routing Problem (MVDRP) [47] where launching a drone along an edge is allowed. Both discretized and continuous methods will be used. Analyses will be conducted which quantifies potential gains by using edge launches. Our approach can be applied to other truck-and-drone problems since it is easily generalizable.

This chapter is organized as follows: Section 5.2 contains an overview of the related literature meaningful to the proposed study. In Section 5.3, the original MVDRP problem description and solution method, proposed in [47], called Route, Transform and Shortest Path (RTS), are reviewed. Section 5.4 first defines the MVDRP with Edge Launch (MVDRP+EL), then describes three procedures which improve solution quality. The results of extensive experimentation utilizing various combinations of the improvement methods are reported in Section 5.5. Finally, Section 5.6 presents conclusions and perspectives on future work on this topic.

5.2 Related works

In this section, we provide an overview of routing optimization problems which consider the use of hybrid truck-and-drone delivery schemes.

The first routing optimization problems related to the use of a truck and a drone for parcel delivery were presented in [43]. The authors introduced two original problems where a vehicle and a drone, starting from a depot, must serve a set of customers. In both problems, a customer can be served by the drone or by the vehicle, and the objective is to minimize the time needed by them to return to the depot after all the customers have been served. In the first problem, titled the Flying Sidekick Traveling Salesman Problem (FSTSP),

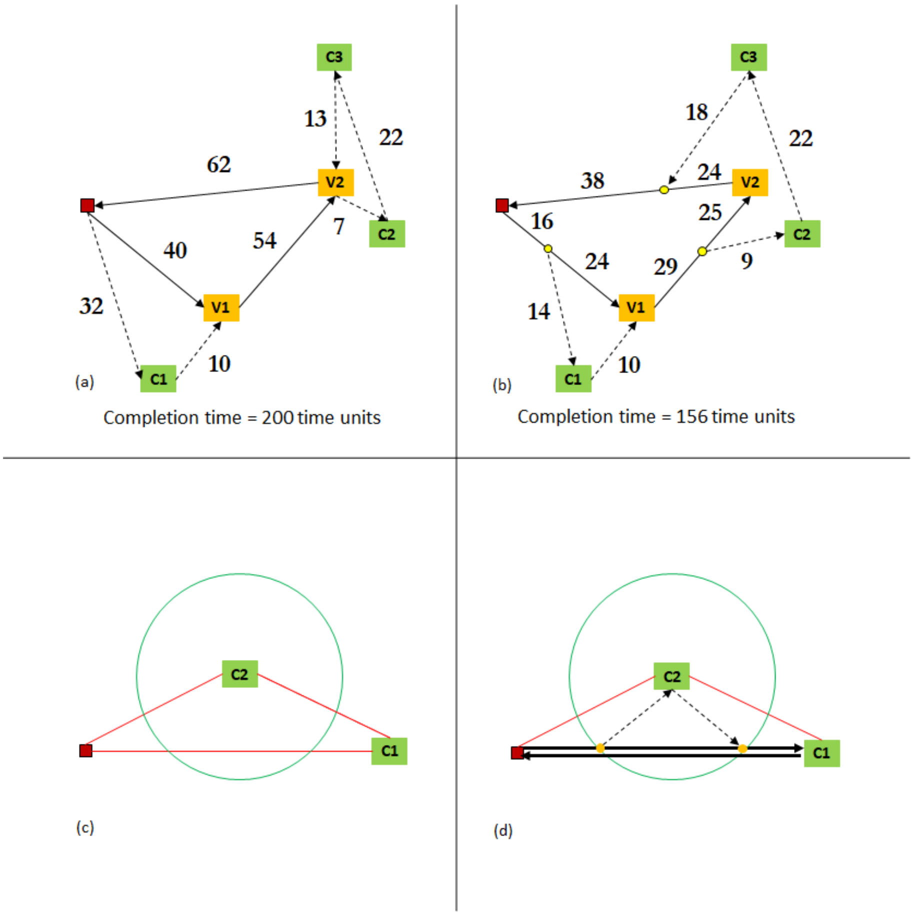


Figure 5.1: A comparison between the solution of the proposed approach and the traditional schemes.

the vehicle may directly serve customers or act as a mobile depot and battery swap station for the drone, which is capable of carrying one homogeneous package to a customer. They also introduced the Parallel Drone Scheduling TSP (PDSTSP), where the drone delivers packages that are within flight range to the depot. The authors proposed two Mixed-Integer Linear Programming (MILP) formulations for the problems and two heuristics since only instances up to 10 customers can be solved to optimality within a reasonable amount of time.

A similar problem called the Traveling Salesman Problem with Drone (TSP-D) was proposed independently in [1]. The main difference is that, in the TSP-D. a node can be visited more than once. The authors proposed an Integer Programming (IP) model and several route-first, cluster-second heuristics based on local search and dynamic programming. The same authors proposed a dynamic programming based exact method for the TSP-D in [8].

In [32], the authors define a min-cost version of the TSP-D where the objective is to minimize total operational costs, not just route duration. The authors proposed a MILP formulation for the problem similar to the one proposed by Murray and Chu for the FSTSP in [43]. A Greedy Randomized Adaptive Search Procedure (GRASP) heuristic and an adaptation of the heuristic in [43] for the FSTSP were proposed to solve larger instances.

A two-stage iterative algorithm was proposed to solve the TSP-D in [63]. In the first stage, a set of truck routes that visit a subset of customers are generated to determine a lower bound for TSP-D solutions that use the same truck routes and serve the other customers by drone. In the second stage, for each truck-route, the drone route to serve the remaining customers is determined solving a MILP model which minimizes the truck-waiting time. The procedure is repeated until the best lower bound determined in the first stage is lower than the incumbent TSP-D solution determined.

A variant of the TSP-D, named the Horsefly Routing Problem (HRP), inspired by commercial systems developed by AMP Electric Vehicles and the University of Cincinnati, was presented in [13]. In the HRP, the truck acts exclusively as a mobile depot for the drone which is solely responsible for the last-leg of delivery. The authors analytically determine completion time upper and lower bounds under the assumption that demand is continuously distributed in the Euclidean plane. Moreover, they characterize the savings obtainable using the HRP as a function of truck and UAV speed.

A problem which focuses exclusively on the optimal determination of the drone route was presented in [10]. The authors defined the Drone Scheduling

Problem (DSP) and proposed two MILP formulations to model it. The problem objective is to determine the drone schedule which minimizes the completion time given a predetermined truck route.

The first paper to consider the use of multiple vehicles equipped with drones is [62]. The authors define the Vehicle Routing Problem with Drones (VRPD) as the problem where a set of customers has to be served by a homogeneous fleet of trucks, each carrying a certain number of homogeneous drones. The problem objective is to minimize the completion time. In VRPD, a drone flight is feasible if the duration of the flight is less than the maximum battery duration of the drone and the drone lands on the same truck from which it launched. The authors studied the problem from a worst-case perspective to determine the maximum potential savings obtainable by the use of this delivery system compared to traditional ones. The same authors extended their work in [48] and consider the impact on objective value of the battery capacity of the drone and various distance metrics for the trucks and drones. Moreover, the authors point out that the VRPD may be characterized as an intermediate problem between the vehicle routing problem (VRP) and the close-enough vehicle routing problem (CEVRP).

Two heuristic algorithms were proposed in [52] for the VRPD. The two heuristics are based on the route-first, cluster-second framework. The first is a two-stage heuristic that, in the first stage, determines a VRP solution by exclusively considering the fleet of trucks and, in the second stage, improves the solution by inserting drones to serve some customers. The second heuristic first determines a TSP solution. Then, the solution is improved allowing the drones to serve some customers. Finally, the route is split into different subroutes, where one truck is assigned to each subroute.

The same authors proposed a new MILP formulation for the VRPD in [51]. Moreover, the authors provided an extensive literature review on hybrid truck-and-drone problems and derived some valid inequalities for the model. Finally, an original Variable Neighbourhood Search procedure is presented to tackle large instances.

A variant of the VRPD is introduced in [49]. The authors present a MILP formulation of a VRPD with time windows where the objective is minimizing the total route length. The formulation is solved using a commercial optimization solver and tested on instances up to 20 customers.

The only other paper to consider the possibility of edge launches is [53]. The authors define the Vehicle Routing Problem with Drones and En Route Operations (VRPDERO). The problem is similar to the VRPD with the addition of the ability to launch and retrieve the drone at some discrete points that are located on each arc. The authors proposed a MILP formulation to solve the problem for instances up to 10 customers and a hybrid Variable Neighborhood Search/Tabu Search algorithm for instances up to 50 customers.

5.3 Multi-visit Drone Routing Problem: Problem Description and Summary of Previous Solution Method

In this section, we describe the Multi-visit Drone Routing Problem. Then we review the Route, Transform and Shortest Path algorithm proposed in [47] to determine a heuristic solution for the MVDRP. This algorithm forms the basis for the proposed method further discussed in Section 5.4.

5.3.1 MVDRP formulation

The Multi-visit Drone Routing Problem (MVDRP) is a scheme of delivery which considers a tandem between a truck and a drone to serve a set of customers. The tandem is initially located at a warehouse, called the depot. The objective of the MVDRP is to minimize the completion time, which is the time needed for the tandem to serve all the customers and return to the depot. An overhead time, called $LaunchP$, is a penalty which arises each time the truck launches the drone. Additional characteristics and assumptions of the truck and drone in MVDRP follow.

Truck - characteristics and assumptions. The truck can serve customers or act as a mobile depot and recharging location for the drone. However, when the drone is airborne, the truck must proceed directly to the rendezvous location and may not service customers. The route of the truck is constrained by the road network. The truck has infinite capacity and range.

Drone - characteristics and assumptions. The drone launches from the truck to make deliveries to one or more customers and then returns to the truck for recharging (or battery swap) and to pick up further packages for the successive launches. The drone route is not constrained by the road network, hence it can move straight from one location to another. The drone speed is known in advance and it is usually greater than truck speed. w_{max} is the maximum weight that the drone can carry. Finally, the drone is equipped with a battery which has a maximum energy capacity e_{max}. The energy dissipation rate by the drone per unit of time is a function of the sum of package weights carried. The drone must return to the truck before the battery is depleted.

5.3.2 RTS Algorithm

Due to the difficulty in finding an optimal solution to the MVDRP, the heuristic RTS algorithm was proposed. The RTS algorithm consists of three steps. In the first step, called *Route*, the customer visit order is determined by solving a Traveling Salesman Problem (TSP) over the customers. Given a set of customers, C, a TSP is solved on C where each customer is connected to each other customer by an edge whose weight is equal to the time needed by a drone to go from one customer to the other one. The output of this step is a sequence of customers $\{c_1, ..., c_{|C|}\}$ which indicates the fixed order according to which the customers will be served in the final solution.

In the second step, called *Transform*, a graph G' is constructed to represent all the possible routes for the tandem given the customer visit order determined in the first step. This step is the fundamental one of the algorithm. It is based on the concept of an *operation*. An operation for the RTS is a movement of the tandem such that at the beginning and at the end of the operation the drone and truck are together in the same locations. Its expression consists of a 4-tuple (i_1, j_1, i_2, j_2) which means that the operation begins with the tandem located at the vertex i_1 in a state where the first j_1 customers have been serviced and ends with the tandem at the vertex i_2 in a state where the first j_2 customers have been serviced. This expression also provides information about truck and drone routes within the operation. Indeed, we can retrieve this information on the basis of i_1, j_1, i_2, j_2 as follows:

- If $i_1 \neq i_2 \wedge j_1 = j_2$, then the truck carried the drone from i_1 to i_2. Because $j_1 = j_2$, no customers were serviced.

- If $i_1 = i_2 \wedge j_1 < j_2$, then the truck stops at i_1 and the customers in $[c_{j_1+1}, ..., c_{j_2}]$ are served. If the position of the served customers is equal to the position of i_1, then the truck serves these customers. Otherwise, the drone makes the deliveries moving from customer to customer while the truck waits at i_1.

- If $i_1 \neq i_2 \wedge j_1 < j_2$, then the truck launches the drone at i_1 and moves to the rendezvous location at i_2. The drone serves all the customers in $[c_{j_1+1}, ..., c_{j_2}]$.

It is also possible to compute the maximum weight carried by a drone during an operation, which is its weight at take-off. Let w_c be the weight of the package demanded by customer c. The maximum weight carried by a drone within the operation (i_1, j_1, i_2, j_2) is given by the following expression:

$$w(i_1, j_1, i_2, j_2) = \sum_{k=j_1+1}^{j_2} w_k \ .$$

Moreover, the weight carried by the drone after servicing customer c_a but before arriving to customer c_{a+1}, such that $j_1 + 1 \leq a \leq j_2$, may be computed as $w(i_1, j_1, i_2, j_2) - \sum_{k=j_1+1}^{a} w_k$. That is, the take-off weight minus the sum of the weights of packages already delivered within the operation.

The time needed to perform the operation (i_1, j_1, i_2, j_2) may be expressed as:

$$t(i_1, j_1, i_2, j_2) = \max\{t_d(i_1, j_1, i_2, j_2), t_t(i_1, j_1, i_2, j_2)\} + I * LaunchP$$

where:

$$t_t(i_1, j_1, i_2, j_2) = d_t(i_1, i_2)/Speed_t,$$
$$t_d(i_1, j_1, i_2, j_2) = (d_d(i_1, c_{j_1+1}) + \sum_{x=j_1+1}^{j_2-1} d_d(c_x, c_{x+1}) + d_d(c_{j_2}, i_2))/Speed_d,$$

t_t represents the amount of time required for the truck to travel from i_1 to i_2. t_d represents the amount of time for the drone to go from launch location i_1 , service customers $[c_{j_1+1}, ..., c_{j_2}]$ (in order), then return to the truck at i_2. If $t_d > 0$ then indicator variable I takes value 1, which effectively imposes the overhead launch penalty; if $t_d = 0$, then $I = 0$ and no launch penalty is assessed. The functions $d_t(i_1, i_2)$ and $d_d(i_1, i_2)$ output the distance which has to be covered by a truck and a drone to go from i_1 to i_2, respectively. $Speed_t(Speed_d)$ is the speed of the truck (drone).

Because we can compute the weight of packages carried and the duration of each flight segment of the drone within an operation, we can also compute the total energy expenditure. Let $e(i_1, i_2, w)$ be the function which expresses the amount of energy spent by a drone moving from i_1 to i_2 carrying a weight equal to w and let h be a constant that indicates the rate of energy consumption per unit time for a drone, whenever it is hovering. The energy expenditure (ee)

of an operation (i_1, j_1, i_2, j_2) can be computed as follows:

$$ee(i_1, j_1, i_2, j_2) = e(i_1, c_{j_1+1}, \sum_{k=j_1+1}^{j_2} w_k) + \sum_{x=j_1+1}^{j_2-1} e(c_x, c_{x+1}, \sum_{k=x+1}^{j_2-1} w_k) +$$
$$e(c_{j_2}, i_2, 0) + h * \max\{0, t_t(i_1, j_1, i_2, j_2) - t_d(i_1, j_1, i_2, j_2)\} \ .$$

The first three terms in the energy expenditure expression represent the energy required by the drone to move from the launch position to the first customer, to go from one customer to the next in $[c_{j_1+1}, ..., c_{j_2}]$, and to move from the last customer to the landing position, respectively. The fourth term expresses the energy spent by the drone if it must wait for the truck at the landing position. Indeed, the drone cannot land if the truck has not yet arrived, so it must hover until the truck has reached the rendezvous position.

Once the concept of operation has been defined, it is possible to explain the *Transform* phase of the algorithm, where graph G' is constructed. This process takes as input the vertices of the original network $G = (V, E)$ representing the road network and generates $|C| + 1$ copies for each vertex in V. In particular, for a vertex i in V we will generate the vertices $\{(i, 0), (i, 1), ..., (i, |C| - 1), (i, |C|)\}$ in V' . In this way, we are linking each vertex copy to a different number of served customers. Hence, a vertex of G' is denoted by $v'_{i,j} \in V'$ to represent the original vertex $i \in V$ in a state where customers $c_1, c_2, ..., c_j$ have been serviced. Thus, an operation (i_1, j_1, i_2, j_2) may be thought of as an arc in G', linking v'_{i_1,j_1} to v'_{i_2,j_2}. The cost of the arc $(v'_{i_1,j_1}, v'_{i_2,j_2})$ is equal to the time needed to perform the corresponding operation, $t(i_1, j_1, i_2, j_2)$, if the operation is feasible. Considering the weight and energy constraints, it is possible to formally define the arc set (i.e., the set of feasible operations), A', of the new graph G' as follows:

$$A' = \{(v'_{i_1,j_1}, v'_{i_2,j_2}) : (j_1 \leq j_2) \wedge ee(i_1, j_1, i_2, j_2) \leq e_{max} \wedge w(i_1, j_1, i_2, j_2) \leq w_{max}\} \ .$$

Therefore, at the end of the second phase, a directed graph $G' = (V', A')$ is built, in which the cardinality of V' is equal to $|V| * (|C| + 1)$.

Finally, in the last step of RTS (*Shortest Path*), the heuristic solution to the MVDRP is determined by solving a Shortest Path Problem on the graph G'. Indeed, once the graph G' has been constructed, it is possible to determine a MVDRP solution solving a shortest path problem on G' where the origin is the vertex corresponding to the depot with no customer served $(v'_{depot,0})$ and the destination is the vertex corresponding to the depot with every customer served $(v'_{depot,|C|+1})$. An example of a MVDRP solution on G' is shown in the

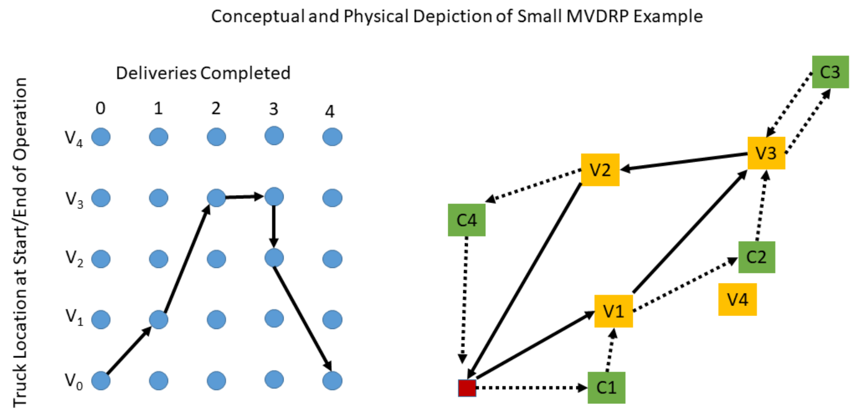


Figure 5.2: An example of the representation of a solution on the graph G' and on the original graph G on the left and on the right, respectively.

left half of Figure 5.2. Each row corresponds with some launch/landing location. Each column corresponds to a number of completed deliveries. Arrows connect the start and end location and state of each operation used by the tandem. On the right of Figure 5.2, is reported the associated representation of the physical path of the solution. The red square is the depot location. Green and yellow squares represent locations of customers and possible launch/landing sites, respectively. Solid black lines show the path of the truck; dashed black lines show the path of the drone.

5.4 MVDRP+EL: Definition and Proposed Solution Method

5.4.1 Defining MVDRP+EL

In MVDRP, we made the assumption that the set of feasible locations where a drone may be launched or retrieved was given by a discrete set V. We define MVDRP with Edge Launch (MVDRP+EL) identically, except that a drone is allowed to launch from/rendezvous with the truck at any location $v \in V$ or at any location along an edge $e \in E$.

5.4.2 Solution Framework to MVDRP+EL

The original RTS algorithm assumes that all launch/rendezvous locations were from the discrete set of locations V. In the case that the optimal customer visit order is known, RTS produces the optimal MVDRP solution. Our heuristic solution framework for MVDRP+EL involves an iterative improvement procedure where the RTS algorithm is used as a subprocedure.

In principle, if we know the optimal customer visit order and the RTS algorithm is executed on a graph $G^*(V^*, E^*)$ where V^* is the set of all points along all edges in the road network and E^* is the related edge set, it would be possible to determine the optimal solution of the MVDRP+EL. However, as V^* and E^* are uncountably infinite, this is clearly not practical from a computational point of view.

We could consider, however, a finite discretization of each edge $e \in E$ and then apply RTS. Indeed, we recall that, using the RTS, from a graph G with $|C|$ customers and n vertices builds a graph G' with $n * (|C| + 1)$ vertices. Suppose we introduce k additional launch/retrieval locations along each edge of the graph, subdividing each edge into $k + 1$ smaller segments. The number of vertices of G would be equal to $n + k|E|$ and then the number of vertices of G' would be equal to $(n + k|E|) * (|C| + 1)$. To preserve computational tractability, the number of discretized points added per edge, must be limited.

Based on this, we propose a method based on three improvement ideas applied to the RTS algorithm. Two ideas are devoted to determining additional launch/retrieval location points. The first one iteratively discretizes the road network edges. The second one assumes a fixed truck route and set of drone operations as input, but optimizes launch and retrieval locations for each operation in a continuous manner. Finally, the last improvement is related to the customer visit order and is devoted to determine different orders by varying the metric considered solving the TSP in the first step of the RTS.

Improvement 1: Iterative Discretization of Road Network Edges

This improvement iteratively forms discretizations of the edges of graph G. The total number of intermediate points added to the original graph in each iteration is fixed and it is equal to the parameter ip_{max}. We limit the number of intermediate points because the storage of G', particularly arc set A', may run into memory issues. Because ip_{max} is limited, we wish to strategically budget our use of intermediate points across the graph. A parameter $pot_e(i)$ which indicates the potential number of intermediate points to be generated on edge e is computed at each iteration, i, of the discretization procedure. The analytic expression of $pot_e(i)$ is the following:

$$pot_e(i) = e^{\{[(-f(e)/T(i)]+l_e/l_{max})\}}$$

where

- l_e is the length of the edge e;
- l_{max} is the maximum length of any edge, computed as $max_{e \in E}(l_e)$;
- $f(e)$ is a measure of the distance of e from the last solution;
- $T(i)$ is the temperature of iteration i, which is a decreasing function.

In the expression of $pot_e(i)$, we consider the length of each edge. Introducing an intermediate point on a very short edge is unlikely to introduce new feasible operations that substantially reduce completion times, because the edge end points are not far from any choice of an intermediate point. Because the scale of distances involved is larger for longer edges, discretization of longer edges may have a greater impact on objective values. Therefore, we want to generate more intermediate points for the longer edges, rather than the shorter ones.

The term $f(e)$ measures the distance of an edge e from the truck route in the solution of the previous iteration. In the first iteration, $f(e) = \infty, \forall e \in E$. The idea is that if an edge e is near the truck route of the most recent solution (i.e., $f(e)$ is small), we would like to prioritize this edge for more intermediate points, as this edge is a good candidate to be part of the final solution. Hence, we want to generate more intermediate points for these edges rather than for the peripheral edges, far away from the existing solution route.

The distance $f(e)$ is equal to the minimum distance from one of the endpoints of e to one of the vertices belonging to the truck route of the last solution found. Let E_t be the set of the edges traversed by the truck in the last solution found, and let fp_e and sp_e be the endpoints of the edge e, then the formal expression of $f(e)$ is the following:

$$f(e) = \min_{j \in E_t} \{d_t(fp_e, fp_j), d_t(fp_e, sp_j), d_t(sp_e, fp_j), d_t(sp_e, sp_j)\}$$

where d_t indicates the truck distance between two vertices.

Finally, we have introduced in the expression of $pot_e(i)$ the parameter $T(i)$, which we refer to as the temperature, inspired by simulated annealing approaches. The temperature parameter changes according to a simple cooling schedule. At each iteration, the temperature is decreased by multiplying by a constant β. In particular, $T(i+1) = \beta * T(i)$, where β is in $[0, 1]$. As the number of iterations, i, increases, $T(i)$ decreases in value (cools). Thus, the relative importance of $f(e)$ in the computation of $pot_e(i)$ increases. That is, in early iterations, when we are determining $pot_e(i)$, less emphasis is given to the proximity of an edge to the previous solution; as the solution cools, we begin to focus more on edges near to the solution. As the procedure progresses through iterations and the temperature cools, the focus gradually shifts from a more equitable discretization of edges across the whole network to a focused refinement of edges that are near the incumbent solution. In other words, our strategy changes gradually from global search to local search.

Once the value of $pot_e(i)$ is determined for every edge, the parameter $ip_e(i)$, which represents the actual number of intermediate points to be allocated to edge e in iteration i, is computed by taking into account ip_{max} and $pot_e(i)$ of all the edges.

At each iteration i, our discretization improvement procedure may be summarized by the following steps:

1. Compute $pot_e(i)$ for each edge e.
2. Compute $totp(i)$ as the sum of the potential number of intermediate points of each edge, $totp(i) = \sum_{e \in E} pot_e(i)$.
3. Compute $ip_e(i)$ for each edge e, $ip_e(i) = pot_e(i) * ip_{max}/totp(i)$.
4. Sort edges according the value of $ip_e(i)$ in descending order.
5. Move through the list of sorted edges, and generate $\lceil ip_e(i) \rceil$ intermediate points for edge e, where $\lceil . \rceil$ is the ceiling function. These intermediate points are uniformly spaced along the edge. Continue allocating intermediate points in this manner until ip_{max} intermediate points have been generated.
6. Run the RTS algorithm on the resulting graph.

The iterative process ends when two consecutive iterations produce the same truck route or if the temperature is lower than a threshold T_{min}. We point out that at the beginning of each iteration the intermediate points previously generated are removed and the new ones will be added to the original graph G. In each iteration, we have $n + ip_{max}$ vertices in V'. Moreover, we highlight that if the truck cannot use edge launches on some roads then it is possible to take this situation easily into account just imposing that $ip_e(i) = 0$ for these edges. An example of the iterative discretization of road network edges is reported in Figure 5.3, using the same notation of Figure 5.2 with the addition of red lines and yellow circles to indicate the road network edges and the intermediate points, respectively. In particular, Figures 5.3(a) and 5.3(b) display the network with the addition of the intermediate points when $T = T_0$ and the corresponding solution, respectively. Then, the Figures 5.3(c)-5.3(d) and 5.3(e)-5.3(f) show the discretized network-solution for decreasing values of T. We can observe that in the first iteration the intermediate points are generated on the longest edges. As T decreases, the intermediate points are more often added to edges on or near the last incumbent solution.

Improvement 2: Continuous Optimization of Launch/Retrieval Locations

In this improvement procedure, we assume that we begin with a feasible MVDRP+EL solution. We fix the truck route and fix the ordered groups of customers that are serviced in each of the operations. What remains fluid, however, is where the drone will launch/rendezvous for each operation. We will seek to optimize the location of the launch and rendezvous for each operation, subject to the restriction that the truck route is fixed and the ordering of drone operations must be preserved.

The truck route time plus mandatory launch penalties represents a lower bound to the solution objective function value. Indeed, if the drone is perfectly synchronized with the truck along the route, then the truck will not incur additional waiting time for the drone. If the truck has to wait for the drone to arrive at a rendezvous location, then this waiting time delays the completion of the route. We will determine the set of launch and rendezvous locations which minimizes the waiting time of the truck and, therefore, minimizes completion time. To this end, we have developed a Mixed Integer Second Order Cone Programming (MISOCP) model. In order to provide the model formulation, let us introduce the following notation:

Known inputs :

- te - number of edges traversed by the truck throughout its route.
- S - Sequence of vertices in the truck route, $S = \{v_{s_0}, v_{s_1}, v_{s_2}, ..., v_{s_{te}}\}$.
- E_t - Sequence of edges in the truck route, $E_t = \{e_1, e_2, ..., e_{te}\}$.
- L_t - edge lengths in the truck route, $L_t = \{l(e_1), l(e_2), ..., l(e_{te})\}$.

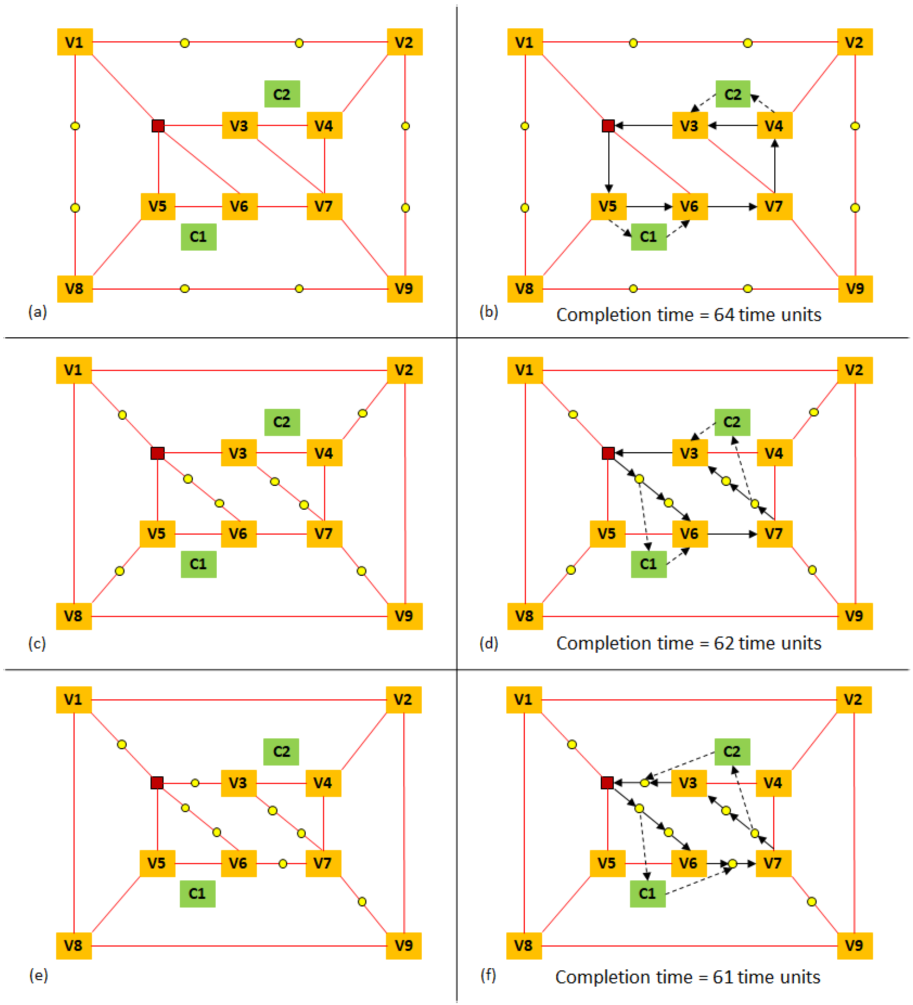


Figure 5.3: An example of iterative discretization of road network edges

- Customer locations - $c_1, c_2, ..., c_{|C|}$.
- O - set of the operations along the route that include a drone flight. $O = \{o_1, o_2, .., o_{|O|}\}$.
- $C(o)$ - set of the customers served within operation o, for each $o \in O$.
- $w(o)$ - sum of the package weights of the customer in $C(o)$.
- DL - Location of the depot.
- ver_e - unit vector which expresses the direction of the edge e.
- $Speed_d$ - speed of the drone.
- $Speed_t$ - speed of the truck.
- $e(w*)$ - function which expresses the instantaneous drone energy consumption carrying a weight equal to w*.
- h - function which expresses the instantaneous drone energy consumption to hover.

Preprocessed parameters :

- $mindist_c$ - minimum distance between the customer c and any edge of the truck route, $mindist_c = \min_{e_i, i=1,..tn} dist(c, e_i)$.
- $earliest_o$ - the earliest location encountered along the truck route that could feasibly serve as the launching location for operation o.
- $latest_o$ - the latest location encountered along the truck route that could feasibly serve as the landing location for operation o.
- $tintra_d^o$ - time spent by the drone from the arrival at the first customer of the operation o until depearture from the last customer of the operation o.
- $eintra_d^o$ - energy spent by the drone from the arrival at the first customer of the operation o until departure from the last customer of the operation o.

The $earliest_o$ and $latest_o$ parameters are determined by using the energy constraint. Indeed, the earliest launching location for each operation o, $earliest_o$ is the first location X encountered along the truck route such that the following inequality holds:

$$e(W(o)) * d_d(X, fc(o)) + eintra_d(o) + e(0) * mindist_{lc(o)} \leq e_{max}$$

where $fc(o)(lc(o))$ is the first (last) customer served within the operation o (i.e., the first (last) element of the ordered set $C(o)$). Similarly, $latest_o$ is the last location X encountered along the truck route such that the following inequality holds:

$$e(W(o)) * mindistfc(o) + eintra_d(o) + e(0) * d_d(lc(o), X) \leq e_{max} \; .$$

Decision variables :

- $y^o_{e,l}$ - binary variable that is equal to 1 if the truck has completely traversed edge e before launching the drone to perform operation o, 0 otherwise.
- $y^o_{e,r}$ - binary variable that is equal to 1 if the truck has completely traversed edge e before the drone lands after operation o, 0 otherwise.
- s^o_l - integer variable indicating the number of edges completely traversed by the truck before the launch of the drone in operation o.
- s^o_r - integer variable indicating the number of edges completely traversed by the truck before the landing of the drone within operation o.
- $x^o_{e,l}$ - continuous variable in $[0, 1]$ which indicates the portion of the edge e traversed by the truck before the launch of the operation o occurs at.
- $x^o_{e,r}$ - continuous variable in $[0, 1]$ which indicates the portion of the edge e traversed by the truck before the landing of the operation o occurs at.
- loc^o_l - position of the launching location of the operation o.
- loc^o_r - position of the retrieving location of the operation o.
- t^o_d - time needed by the drone to perform operation o.
- $tout^o_d$ - outbound time needed by the drone to move from the launch location to the first customer of the operation o.

- tin_d^o - inbound time needed by the drone to move from the last customer to the retrieval location of the operation o.
- t_t^o - time needed by the truck to perform operation o. This is equal to the driving time from loc_l^o to loc_r^o along the truck route.
- $twait_t^o$ - waiting time of the truck for the drone in operation o.
- $eout_d^o$ - outbound energy spent by the drone to move from the launch location to the first customer of the operation o.
- ein_d^o - inbound energy spent by the drone to move from the last customer to the retrieval location of the operation o.
- eh_d^o - energy spent by the drone to hover, while waiting for the truck at the rendezvous location, in operation o.

On the basis of this notation, it is possible to introduce the following model formulation, where the **objective function** is:

$$\texttt{minimize} \sum_{o \in O} twait_t^o \tag{5.1}$$

where the sum of the truck waiting times is minimized. Indeed, since the truck route is fixed the possible drone waiting time is already included in the truck route duration. On the other hand, the truck waiting time represent an overhead for the truck route duration.

The set of constraints can be divided into 6 subsets:

Consistency constraints

$$t_d^o - t_t^o \leq twait_t^o, \forall o \in O \tag{5.2}$$

$$tout_d^o + tintra_d^o + tin_d^o \leq t_d^o, \forall o \in O \tag{5.3}$$

$$\left(\sum_{e=0}^{te-1} \{[(y_{e,r}^o + x_{e,r}^o) - (y_{e,l}^o + x_{e,l}^o)] * l(e)\}\right)/Speed_t \leq t_t^o, \forall o \in O \tag{5.4}$$

$$\sum_{e=0}^{te-1} y_{e,l}^o = s_l^o, \forall o \in O \tag{5.5}$$

$$\sum_{e=0}^{te-1} y_{e,r}^o = s_r^o, \forall o \in O \tag{5.6}$$

$$loc_r^o = DL + \sum_{e=0}^{te-1} ver_e * (y_{e,r}^o + x_{e,r}^o), \forall o \in O \tag{5.7}$$

$$loc_l^o = DL + \sum_{e=0}^{te-1} ver_e * (y_{e,l}^o + x_{e,l}^o), \forall o \in O \tag{5.8}$$

Constraints (5.2) set the truck waiting time of each operation to be equal to the difference between the time needed for the drone and the truck to perform the operation. Constraints (5.3) define the time needed for the drone to perform each operation. Constraints (5.4) define the truck time needed for each operation as the required driving time from the launch location to the rendezvous location. Constraints (5.5-5.6) set the number of edges completely traversed equal to the sum of them. Constraints (5.7)/(5.8) define the landing/launching position of each operation.

SOC constraints

$$||loc_l^o - loc_{fc}^o||/Speed_d \leq tout_d^o, \forall o \in O \tag{5.9}$$

$$||loc_r^o - loc_{lc}^o||/Speed_d \leq tin_d^o, \forall o \in O \tag{5.10}$$

Constraints (5.9) and (5.10) define for each operation the time needed for the drone to go from the launching position to the first customer location and from the last customer location to the retrieving location, respectively.

Precedence constraints

$$y_{e,l}^o \leq y_{e-1,l}^o, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.11}$$

$$y_{e,r}^o \leq y_{e-1,r}^o, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.12}$$

$$y_{e-1,l}^o - y_{e,l}^o \geq x_{e,l}^o, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.13}$$

$$y_{e-1,r}^o - y_{e,r}^o \geq x_{e,r}^o, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.14}$$

$$s_l^o + \sum_{e=0}^{te-1} x_{e,l}^o \leq s_r^o + \sum_{e=0}^{te-1} x_{e,r}^o, \forall o \in O \tag{5.15}$$

$$s_{o,r} + \sum_{e=0}^{te-1} x_{e,r}^o \leq s_{o+1,l} + \sum_{e=0}^{te-1} x_{e,l}^o, \forall o \in O \tag{5.16}$$

Constraints (5.11)/(5.12) ensure that an edge cannot be completely traversed before the launch/landing of the drone in an operation if the previous edge has not been completely traversed. Constraints (5.13-5.14) guarantee that an edge can only be partially traversed if the previous edge has been completely traversed. Moreover, they also guarantee that an edge cannot be at the same time completely and partially traversed. Constraints (5.15) ensure that the launching position precedes the landing position for each operation. Constraints (5.16) guarantee that the landing position of an operation has to precede the launch location of the successive operation.

Energy constraints

$$eout_d^o = tout_d^o * e(W(o)), \forall o \in O \tag{5.17}$$
$$ein_d^o = tin_d^o * e(0), \forall o \in O \tag{5.18}$$
$$eh^o \geq h * (t_t^o - t_d^o), \forall o \in O \tag{5.19}$$
$$eout_d^o + eintra_d^o + ein_d^o + eh^o \leq e_{max}, \forall o \in O \tag{5.20}$$

Constraints (5.17) and (5.18) define for each operation the energy consumed by the drone to go from the launching position to the first customer location and from the last customer location to the landing position, respectively. Constraints (5.19) define the energy needed by the drone to hover while it is waiting for the truck to arrive. Constraints (5.20) limit the drone energy consumption during a single operation to e_{max}.

Endurance constraints

$$earliest_o \leq s_{o,l} + \sum_{e=0}^{te} x_{e,l}^o, \forall o \in O \tag{5.21}$$
$$s_{o,r} + \sum_{e=0}^{te-1} x_{e,r}^o \leq latest_o, \forall o \in O \tag{5.22}$$

Constraints (5.21/5.22) guarantee that a launch/retrieval location cannot be previous/successive to the earliest launch/latest retrieval one. These constraints are valid cuts in the interest of computational performance based on drone endurance, but are not necessary to maintain feasibility.

Variable definition constraints

$$eh^o \geq 0, \forall o \in O \tag{5.23}$$
$$0 \leq twait_t^o, \forall o \in O \tag{5.24}$$

$$y^o_{-1,l} = 1, \forall o \in O \tag{5.25}$$
$$y^o_{-1,r} = 1, \forall o \in O \tag{5.26}$$
$$y^o_{e,l} \in \{0,1\}, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.27}$$
$$y^o_{e,r} \in \{0,1\}, \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.28}$$
$$s^o_l \in \mathbb{N}, \forall o \in O \tag{5.29}$$
$$s^{|O|}_l = te \tag{5.30}$$
$$d^o_r \in \mathbb{N}, \forall o \in O \tag{5.31}$$
$$x^{|O|}_{e,l} = 0, \forall e \in \{0, ..., te-1\} \tag{5.32}$$
$$x^o_{e,l} \in [0,1], \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.33}$$
$$x^o_{e,r} \in [0,1], \forall o \in O, \forall e \in \{0, ..., te-1\} \tag{5.34}$$

Constraints (5.23) to (5.34) generally relate to the domain of variables (e.g., constraints related to non-negativity, binary/integer nature of variables, special treatment of variables related to depot location).Moreover, we underscore that if it is not allowed to have a drone launch/retrieval on some of the truck route edges then we can take into account this constraint just by changing the nature of the corresponding variables $x^o_{e,l}$ and $x^o_{e,r}$ from continuous in $[0,1]$ to binary. An example of the continuous optimization of launch/retrieval locations is reported in Figure 5.4. In particular, Figure 5.4(a) shows the solution obtained using the RTS algorithm. We can observe that in this solution the truck has to wait for the drone while it it serving the customers. Figure 5.4(b) shows the optimized solution obtained solving the MISOCP. The notation used in the picture is the same used in Figure 5.3.

Improvement 3: Varying TSP Metric to Produce Different Customer Visit Orders

In Section 5.3.2, we showed how the customer visit order is determined by the RTS algorithm as the solution to the Euclidean TSP on the customer set. By fixing the customer visit order, our solution space is restricted, but the size of G' is also limited. Indeed, if we would like to consider all the possible customer visit orders for a graph with n vertices and c customers, it would require us to build a graph G' with $n * (c! + 1)$ vertices.

Rather than massively expanding the size of G', we generate a number of different promising customer visit orders. To this end, we generate various distance metrics, solve a TSP over the customer set for the given distance met-

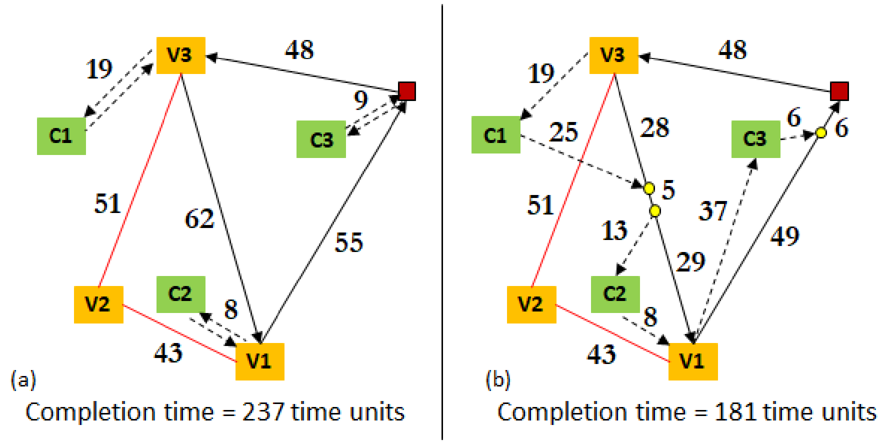


Figure 5.4: An example of continuous optimization of launch/retrieval locations

ric, and fix the resulting customer visit order. As a result, our graph G' only considers operations (arcs) where this visit order is obeyed.

The idea of introducing a new metric distance arises from the kind of distance used in the basic RTS. Indeed, we recall that the customer visit order in the basic RTS is determined using the drone (Euclidean) distance between each customer pair. Although it is reasonable to incorporate the distance metric of the drone, this distance does not take into account the truck travel times. Therefore, we would like to combine the truck and drone distance metrics in some way.

Because the truck cannot always reach every customer location (as the sets V and C are not defined identically), defining the truck distance between two customers requires some adaptation. Thus, each customer c is mapped to the nearest vertex to c that the truck can reach: $near(c) = \min_{i \in V}\{dd(i, c)\}$. Hence the truck distance between two customers c_1 and c_2 will be given by the following expression $d_t(c_1, c_2) = d_t(near(c_1), near(c_2))$. Therefore, it is possible to define a new distance metric as a function of the parameter α, which weights the truck and drone distances. The expression is the following:

$$d(c_1, c_2, \alpha) = \alpha * d_d(c_1, c_2) + (1 - \alpha) * d_t(near(c_1), near(c_2)) \ .$$

The value of α can vary in the range $[0, 1]$ and, at each value, can correspond to a different distance metric d. By varying the distance metric d and solving the associated TSP over the customer set, we encounter a variety of different customer visit orders, each corresponding to a different structure for the graph G'.

5.4.3 Proposed Solution Method

The proposed approach integrates the three improvements previously described with the RTS algorithm in an iterative two-phase solution method. In the outer loop of our solution method, we iteratively increase the value of α. We start with $\alpha = 0$ and increment the value of α by α_{step} after the completion of each loop, until $\alpha > 1$.

Within each loop, we begin by solving a TSP for the given value of α to determine the customer visit order. In the first improvement phase, we apply the iterative edge discretization procedure. The resulting solution is further improved through the continuous improvement procedure, the second phase, which determines the optimal launch and retrieval locations. After the second phase improvement is complete, the loop is complete.

In the case that two distinct values of α produce the same customer visit order, we do not need to repeat the Phase 1 and Phase 2 improvements, as they are deterministic algorithms, and the results will be the same.

We use the solution corresponding to the lowest objective value encountered at any point during our solution process. The structure of the solution method is illustrated in Figure 5.5.

5.5 Computational results

The proposed approach has been implemented in Python and uses Gurobi 8.1.0 to solve the TSPs and the MISOCPs. The computational experiments were carried out on a computer with an Intel Core i7-4750HQ processor operating at 2.00 GHz, 8 GB RAM, and Windows 10 (64 bit) as the operating system.

Our test set consists of 150 instances generated with the same settings described in [47]. This choice is motivated by the assumption that the sets V and C can be disjoint in the tackled problem, unlike other problems as the FSTSP and the TSP-D. In particular, given a number of customer locations, $|C|$, and a specified number of road network vertices, $|V|$, we randomly generated all customer locations and vertices uniformly within a 100 by 100 square region. The depot location was also randomly generated within the same region.

The truck can move at unit speed only on road network edges (i.e., the

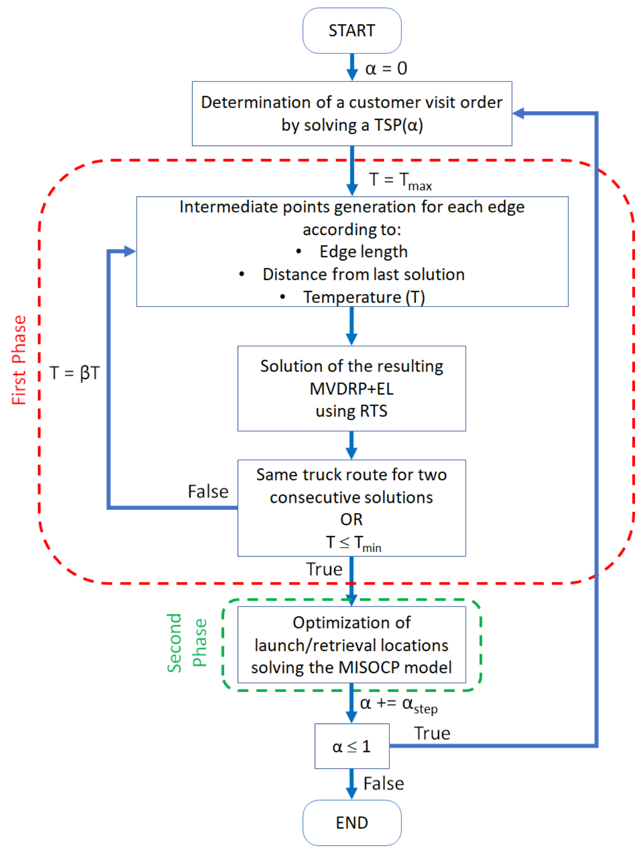


Figure 5.5: The flow-chart of the proposed solution method

edges connecting two road network vertices), characterized by the Euclidean distance. The drone can move from and to any two vertices (customer locations or road vertices) according to the Euclidean distance, but at a speed of 2 units.

The weight of packages, w, demanded by each customer was distributed uniformly over the range [0, 5]. We fixed the energy capacity, e_{max}, to 40 and used $e(w) = (1 + (w/5)^4)$, which is meant to impose a soft cap maximum weight of five pounds, to align with comments made by Amazon CEO, Jeff Bezos, during an interview [5]. The hovering and the launch penalty constants, h and $LaunchP$, were set to 0.5 and to 1, respectively. We set $\alpha_{step} = 0.1$, the initial value of T equal to the length of the longest road network edge, $T_{min} = T_{start}/100$, and β=0.4

We considered instances with $|V|$ equal to 20, 40, 60, 80, and 100. For each value of $|V|$, we considered three values of $|C|$: $0.1|V|, 0.4|V|$, and $0.7|V|$. Then, for a given pair of $|C|$ and $|V|$, we generate two instances with the same customer and road network vertex locations, but we vary the number of road network edges with values of $|E_1|$ and $|E_2|$, with $|E_1| < |E_2|$ to assess the effectiveness of our approach on the basis of the sparsity of the road network. Edges were chosen by starting with a complete graph and randomly deleting edges with a probability of 90% and 75%. Finally, for each combination of $|V|$, $|C|$, and $|E|$, we generated 5 instances. The aggregated results for each value of $|V|$ are reported in this section. The interested reader is referred to the Appendix for the aggregated results for each combination of $|V|$, $|C|$, and $|E|$.

In these computational experiments, the test instances are solved in several ways to assess the improvement obtainable applying each idea independently and combinations of them. Therefore, first, we solved the instances using the RTS algorithm. Then, we solve the instances applying each improvement separately. Successively, we tested each pairwise combination of the improvements. Finally we tested the proposed solution method, which combines all three improvement ideas.

5.5.1 Single Improvement results

In this subsection, we report the results obtained by applying each improvement to the RTS separately. The resulting improvement procedures are the following:

- αRTS - The solution is obtained by solving the RTS for different values of α

- $DRTS1$ - The solution is obtained by applying the first phase of the algorithm with $ip_{max} = |V|/2$.

- $DRTS2$ - The solution is obtained by applying the first phase of the algorithm with $ip_{max} = |V|$.

- $CRTS$ - The solution is obtained by applying the MISOCP to the solution of the basic RTS.

Objective values and computation times found by applying each individual improvement procedure are shown in Tables 5.1 and 5.2, respectively. Table 5.1 shows the percentage savings obtained by applying the single improvements. The cardinality of V is listed in the first column of Table 5.1. The successive columns of Table 5.1 show the average of the percentage savings, compared to the RTS solution, obtained by each improvement. αRTS, $DRTS1$, $DRTS2$, and $CRTS$ indicate the savings computed for each of these three improvement methods, respectively. The final row reports average savings across all instances generated.

From Table 5.1 we observe that the best results are obtained using the improvement $DRTS2$. We also note that the savings averaged only on sparser graphs (i.e., considering the instances with $E_1, |E_1| < |E_2|$) was 4.3% greater than on denser graphs (i.e., considering the instances with $E_2, |E_2| > |E_1|$).

Table 5.2 reports the cardinality of V, and the computation times of the RTS algorithm and of each improvement. We observe that the computation times of the different improvements are greater than that of the RTS. Moreover, we observe that the time required to solve an instance increases with the cardinality of V.

Table 5.1: Savings (%) of single improvement approaches

V	αRTS	$DRTS1$	$DRTS2$	$CRTS$
20	3.03	12.25	13.47	14.25
40	1.28	10.35	12.36	10.29
60	1.19	5.43	7.80	5.84
80	0.45	5.30	6.60	6.59
100	1.06	2.79	3.15	5.55
Mean	1.40	7.22	8.68	8.50

Table 5.2: Times of single improvement approaches

V	RTS	αRTS	$DRTS1$	$DRTS2$	$CRTS$
20	0.07	0.63	1.85	3.40	0.35
40	0.28	2.32	6.62	15.13	0.88
60	0.82	7.09	14.37	38.23	1.77
80	1.82	17.64	26.14	76.47	3.53
100	3.47	34.39	45.22	133.51	5.58
Mean	1.29	12.41	18.84	53.35	2.42

5.5.2 Combined improvement results

We also solved the instances by applying each pairwise combination of the proposed improvements, so obtaining the following procedures:

- $\alpha DRTS1$ - The solution is obtained applying the first phase of the algorithm, with $ip_{max} = |V|/2$, to the customer visit orders obtained using different values for α.

- $\alpha DRTS2$ - The solution is obtained applying the first phase of the algorithm, with $ip_{max} = |V|$, to the customer visit orders obtained using different values for α.

- $\alpha CRTS$ - The solution is obtained applying the MISOCP to the solutions obtained by the basic RTS for different values of α.

- $DCRTS1$ - The solution is obtained applying the MISOCP to the solution obtained by the procedure $DRTS1$.

- $DCRTS2$ - The solution is obtained applying the MISOCP to the solution obtained by the procedure $DRTS2$.

The resulting objective values and computation times from application of pairwise improvements are shown in Tables 5.3 and 5.4, respectively. Table 5.3 shows the cardinality of V in the first column, and the percentage savings (relative to RTS) obtained applying the solution methods resulting from each pairwise combination in the remaining columns.

From Table 5.3 we observe that the best results were obtained using the procedure $DCRTS2$, which reduced objective values by slightly more than 13%, on average. Notably, the two constituent improvements (DRTS2 and CRTS) were the two individual improvements that produced the most savings in Table 5.1. Moreover, the savings obtained applying each combination of improvements is greater than the savings achieved using either constituent improvement process.

Table 5.4 reports the cardinality of V, and the average computation times of procedure for each pairwise improvement combination. First, we observe that the computation times are greater than those of the single improvements. The $\alpha DRTS1$ and $\alpha DRTS2$ are the most time consuming procedures. This behavior can be explained as the DRTS procedure is relatively time consuming, and it is performed for each distinct value of α. Although $\alpha CRTS$, $DCRTS1$, and $DCRTS2$ required less computational time than the other procedures, they produced the greatest amount of savings among the pairs of improvements.

Table 5.3: Saving (%) for combined improvement approaches

V	$\alpha DRTS1$	$\alpha DRTS2$	$\alpha CRTS$	$DCRTS1$	$DCRTS2$
20	15.12	16.36	17.64	17.01	18.63
40	11.58	13.73	11.85	15.67	16.60
60	6.34	8.95	7.47	10.70	12.86
80	3.23	4.57	7.26	8.67	8.77
100	3.75	3.95	6.97	7.77	8.18
Mean	8.00	9.51	10.24	11.96	13.01

Table 5.4: Times for combined approaches

V	$\alpha DRTS1$	$\alpha DRTS2$	$\alpha CRTS$	$DCRTS1$	$DCRTS2$
20	8.57	16.82	2.52	2.25	3.82
40	37.70	84.76	7.13	6.39	13.02
60	95.06	269.88	17.74	15.13	39.71
80	234.82	657.63	36.80	27.21	77.69
100	396.10	1221.32	67.30	46.53	135.01
Mean	154.45	450.08	26.30	19.50	53.85

5.5.3 Solution method results

The results obtained by applying all three components of our proposed solution method are reported in Table 5.5. Consistent with the previous experimentation, we tested two versions of the solution method $\alpha DCRTS1$ and

$\alpha DCRTS2$ with $ip_{max} = |V|/2$ and $ip_{max} = |V|$, respectively. Again, cardinality of V is listed in the first column of Table 5.5. The second and third columns show the average percentage savings, compared to the RTS solution, obtained by the two variants tested. The last two columns show the corresponding average computation times.

We can observe that $\alpha DCRTS2$, which uses a greater number of intermediate points, produced the best objective values with a savings of 14.64% relative to RTS. Naturally, a finer discretization yields a better choice of truck routes and operations in the first phase. The quality of the objective value comes at a cost of additional computational time, however. Indeed, the average computation time of $\alpha DCRTS2$ is three times that of $\alpha DCRTS1$.

Both versions utilizing all three improvement mechanisms show an average savings greater than the ones obtained by any other method tested in the previous experimentation.

Moreover, we highlight that the solution methods using the improvement α can be sped up. Detailed observation of the solutions indicated that the best solution of these methods is generally obtained when α is in the range $[0.6, 1]$. Therefore, by restricting the range of α to $[0.6, 1]$ instead of $[0, 1]$, it is possible to consistently reduce the computation times of these methods without a significant change in objective values.

Table 5.5: Solution method results

	Savings		Time	
V	$\alpha DCRTS1$	$\alpha DCRTS2$	$\alpha DCRTS1$	$\alpha DCRTS2$
20	20.05	21.47	10.82	19.22
40	17.46	18.93	48.15	107.26
60	11.94	13.78	106.14	281.24
80	9.30	9.60	250.13	675.54
100	9.19	9.44	424.24	1250.61
Mean	13.59	14.64	167.90	466.77

5.6 Conclusions

In this study, we introduced a new truck-and-drone routing problem, MV-DRP+EL, which unlike previous papers on this topic in the literature, allows a drone to launch from a truck along an edge. We developed an algorithm that improves the existing RTS heuristic, which was designed for a problem that does not allow edge launches. Utilizing these improvements and the ability to launch along an edge resulted in an average improvement of nearly 15% relative to solutions produced by RTS.

Our algorithm contained three separate improvements that may be summarized as (1) considering various customer visit orders by combining the truck and drone distance metrics, (2) discretizing edges in an iterative fashion to introduce new launch and retrieval points, and (3) using a mixed integer second order cone program to select the best launch and landing location for each operation in a continuous fashion. In addition to testing all three improvements simultaneously, we tested every subset of improvement procedures and found that an iterative discretization of edges was the best individual mechanism for improving solution quality.

Iterative edge discretization and the use of a mixed integer second order cone program are improvements that are only possible when launching along an edge is allowed. The combination of these two improvements yielded more than 13% savings, relative to solutions that only allowed launches to occur at the vertices of the street network. This strongly suggests that the ability to launch along an edge has a non-trivial impact on objective values on truck-and-drone coordination problems.

Future research directions naturally include further refinement of the algorithm and additional testing on other parameter settings. In addition, it may be worth exploring other mechanisms to determine the granularity of discretization of edges, adapt these improvement procedures to slightly different problems, and further study in which contexts edge launch provides the greatest benefit.

Conclusions

In this thesis, a twofold objective has been pursued. The first one is to provide a broad and comprehensive insight on drone systems and the related optimization problems. The second one is to present the main findings and achievements of three studies on truck-and-drone coordination problems.

Drone history, technical properties and applications have been discussed to highlight the current relevance of these systems. Then, some classes of operational problems considering the use of drones have been examined to point out current and possible future research areas. Finally, an extensive overview of the main contributions of the operation research community has been reported to represent the state-of-the-art on this topic.

The first study is concerned about providing a framework able to address a broad swath of truck-and-drone coordination problems. It first transforms the original problem into an equality generalized traveling salesman problem and then into an asymmetric traveling salesman problem. The proposed framework can be easily extended to different variants of this kind of problems. The only requirement consists of the possibility to enumerate the set of feasible operations.

The second study is devoted to the improvement of the current state-of-the-art of the formulations proposed for the Flying Sidekick Traveling Salesman Problem. The findings arising from this study can be extended to other truck-and-drone coordination problem variants to motivate the development of optimal solution method for these problems.

The third study is dedicated to the evaluation of the benefits arising from the possibility of launching and retrieving a drone along the edges of the road network. The performed experimentation strongly suggested that this ability of the truck has a significant impact on truck-and-drone coordination problems in terms of completion time reduction.

Future research directions naturally include improvement of the proposed algorithms. Moreover. it would be interesting to apply the proposed methods to several other problem variants. In any case, since drone technology evolves rapidly, it is very likely that new operational problems with drones can arise so generating the need of the development of new exact or heuristic solution methods.

Appendix

This appendix contains the aggregated results of Chapter 5, for each combination of $|V|$, $|C|$, and $|E|$ averaged over the 5 generated instances. Tables A.1 and A.2 contain the savings and the times solving the instances with each single improvement, respectively. Moreover, Tables A.3 and A.4 show the savings and the times when the instances are solved using each pairwise combination of the improvements. Finally, Table A.5 shows the results from solving the instances through the two versions of the solution method ($ip_{max} = |V|/2$ and $ip_{max} = |V|$).

Table A.1: Savings (%) of single improvement approaches

V	C	E	αRTS	$DRTS1$	$DRTS2$	$CRTS$
20	2	21	0.000	10.711	9.118	13.842
20	2	50	0.000	9.741	12.015	11.276
20	8	21	5.760	11.212	12.228	14.921
20	8	50	3.664	8.514	11.351	15.553
20	14	21	6.033	18.527	18.481	17.419
20	14	50	2.743	14.779	17.628	12.517
40	4	90	0.469	7.567	8.086	1.564
40	4	211	0.004	4.515	6.371	2.292
40	16	90	1.728	18.056	22.162	15.519
40	16	211	2.253	9.831	10.702	12.212
40	28	90	0.805	18.049	17.957	21.834
40	28	211	2.398	4.057	8.872	8.313
60	6	205	0.542	12.921	15.816	2.560
60	6	467	0.000	2.285	2.185	0.665
60	24	205	0.324	7.860	11.074	7.606
60	24	467	0.124	0.273	4.895	4.096
60	42	205	3.978	8.859	9.894	12.993
60	42	467	2.179	0.404	2.930	7.119
80	8	311	0.000	22.517	19.046	6.975
80	8	798	0.095	1.167	2.121	3.400
80	32	311	0.476	6.382	6.952	11.137
80	32	798	0.542	0.261	1.800	4.461
80	56	311	0.876	1.337	8.183	8.859
80	56	798	0.714	0.144	1.515	4.686
100	10	483	1.655	6.698	6.155	5.160
100	10	1245	0.219	1.095	2.170	2.689
100	40	483	2.114	5.901	5.476	7.000
100	40	1245	1.866	0.000	0.000	4.301
100	70	483	0.327	3.031	5.103	8.717
100	70	1245	0.183	0.000	0.000	5.404
Mean			1.402	7.223	8.676	8.503

Table A.2: Times of single improvement approaches

V	C	E	RTS	αRTS	$DRTS1$	$DRTS2$	$CRTS$
20	2	21	0.026	0.196	0.956	2.359	0.139
20	2	50	0.038	0.197	1.201	2.803	0.146
20	8	21	0.063	0.501	1.951	2.879	0.357
20	8	50	0.124	0.928	2.122	3.831	0.353
20	14	21	0.055	0.820	2.260	3.608	0.581
20	14	50	0.127	1.126	2.616	4.914	0.554
40	4	90	0.114	0.616	3.987	8.298	0.327
40	4	211	0.210	0.661	3.453	9.039	0.447
40	16	90	0.230	2.253	7.092	15.415	0.946
40	16	211	0.280	2.010	6.668	14.110	0.708
40	28	90	0.404	4.432	11.162	22.719	1.483
40	28	211	0.441	3.976	7.353	21.185	1.391
60	6	205	0.376	1.173	10.652	25.242	0.677
60	6	467	0.562	1.937	8.852	17.247	0.889
60	24	205	0.708	5.823	15.839	43.401	1.513
60	24	467	0.840	6.261	8.526	32.668	1.679
60	42	205	1.098	12.634	24.850	65.219	2.879
60	42	467	1.341	14.741	17.530	45.595	2.970
80	8	311	0.643	2.925	18.227	42.890	1.209
80	8	798	0.918	2.771	10.112	31.088	1.457
80	32	311	1.680	17.810	37.648	101.450	3.615
80	32	798	2.008	18.738	18.762	47.706	3.764
80	56	311	2.576	30.494	43.808	148.874	5.040
80	56	798	3.120	33.096	28.276	86.810	6.067
100	10	483	1.257	7.287	38.513	94.128	2.189
100	10	1245	1.594	7.318	27.774	64.931	2.648
100	40	483	3.060	31.599	52.536	194.523	4.186
100	40	1245	3.770	32.948	28.710	64.356	4.722
100	70	483	5.026	59.011	79.461	287.512	8.895
100	70	1245	6.113	68.162	44.299	95.618	10.813
Mean			1.293	12.415	18.840	53.347	2.421

Table A.3: Saving (%) for combined improvement approaches

V	C	E	$\alpha DRTS1$	$\alpha DRTS2$	$\alpha CRTS$	$DCRTS1$	$DCRTS2$
20	2	21	10.711	9.118	13.842	13.842	13.157
20	2	50	9.741	12.015	11.276	11.276	14.429
20	8	21	15.789	16.972	21.224	16.873	16.873
20	8	50	14.393	17.154	20.481	11.495	18.208
20	14	21	23.050	23.484	23.989	27.417	27.970
20	14	50	17.046	19.411	15.032	21.183	21.142
40	4	90	7.567	8.086	2.049	8.853	8.853
40	4	211	4.556	6.570	6.360	6.326	9.848
40	16	90	18.290	22.399	15.950	24.169	27.513
40	16	211	11.536	11.874	12.950	14.644	14.129
40	28	90	19.755	21.644	22.736	29.676	25.982
40	28	211	7.768	11.786	11.050	10.326	13.282
60	6	205	12.921	15.816	4.009	15.852	19.331
60	6	467	2.285	3.366	0.665	3.939	5.261
60	24	205	9.601	14.315	9.734	14.583	16.847
60	24	467	2.070	6.893	6.706	5.282	9.054
60	42	205	8.998	10.229	15.753	18.428	18.642
60	42	467	2.179	3.107	7.939	6.093	8.015
80	8	311	5.580	5.320	7.713	9.572	6.737
80	8	798	1.291	2.246	3.400	4.127	4.892
80	32	311	6.693	7.246	11.815	16.268	14.418
80	32	798	0.803	2.342	5.028	5.630	5.960
80	56	311	4.146	8.771	9.872	10.651	13.402
80	56	798	0.858	1.515	5.745	5.774	7.238
100	10	483	8.947	6.779	7.122	8.613	9.638
100	10	1245	1.095	2.501	2.689	3.854	4.476
100	40	483	7.382	7.293	10.192	12.479	12.976
100	40	1245	1.866	1.866	5.688	4.301	4.301
100	70	483	3.031	5.103	8.817	11.993	12.314
100	70	1245	0.183	0.183	7.295	5.404	5.404
Mean			8.004	9.513	10.237	11.964	13.010

Table A.4: Times for combined approaches

V	C	E	$\alpha DRTS1$	$\alpha DRTS2$	$\alpha CRTS$	$DCRTS1$	$DCRTS2$
20	2	21	1.126	2.530	0.309	1.115	2.582
20	2	50	1.359	2.962	0.305	1.381	3.031
20	8	21	8.716	13.190	1.911	2.487	3.358
20	8	50	10.799	23.208	2.932	2.432	4.209
20	14	21	16.253	29.653	5.090	2.832	4.380
20	14	50	13.194	29.364	4.594	3.234	5.369
40	4	90	13.996	29.086	1.375	4.255	8.720
40	4	211	8.497	21.832	1.210	3.741	9.437
40	16	90	58.454	126.956	9.255	7.829	16.210
40	16	211	35.128	79.170	5.545	7.169	14.723
40	28	90	96.943	222.140	20.814	12.128	23.681
40	28	211	47.985	130.886	12.182	8.086	21.929
60	6	205	26.706	62.634	2.099	11.094	25.764
60	6	467	23.251	65.729	3.035	9.286	21.165
60	24	205	130.761	316.645	14.519	16.539	44.195
60	24	467	69.684	241.629	12.276	9.227	33.418
60	42	205	211.636	648.063	40.785	26.002	66.507
60	42	467	108.318	284.590	33.751	18.625	47.196
80	8	311	68.663	181.041	5.014	18.758	43.552
80	8	798	34.227	96.991	4.226	10.697	31.864
80	32	311	380.008	979.385	41.270	38.620	102.648
80	32	798	170.631	513.599	32.746	19.838	48.925
80	56	311	515.811	1555.306	75.659	45.298	150.434
80	56	798	239.562	619.437	61.888	30.066	88.717
100	10	483	187.952	499.387	11.686	39.190	95.028
100	10	1245	83.165	275.734	10.793	28.581	66.029
100	40	483	539.700	1877.957	58.606	53.781	195.901
100	40	1245	242.911	623.361	45.485	29.842	65.701
100	70	483	875.954	3049.412	129.568	81.393	289.471
100	70	1245	446.891	1002.058	147.653	46.413	97.933
Mean			155.609	453.464	26.553	19.665	54.403

Table A.5: Solution method results

			Savings		Time	
V	C	E	$\alpha DCRTS1$	$\alpha DCRTS2$	$\alpha DCRTS1$	$\alpha DCRTS2$
20	2	21	13.842	13.157	1.285	2.753
20	2	50	11.276	14.429	1.540	3.189
20	8	21	22.275	22.275	10.357	15.007
20	8	50	17.988	23.502	12.711	25.412
20	14	21	31.441	32.036	17.987	32.921
20	14	50	23.501	23.393	21.044	36.024
40	4	90	9.339	9.339	14.974	30.450
40	4	211	8.951	10.784	9.207	22.772
40	16	90	24.951	28.177	66.204	135.615
40	16	211	14.838	16.886	32.071	72.800
40	28	90	30.453	29.374	110.302	242.636
40	28	211	16.242	19.008	56.161	139.285
60	6	205	15.852	19.331	28.129	64.118
60	6	467	3.939	5.261	24.718	67.615
60	24	205	16.425	18.399	139.550	326.720
60	24	467	7.852	11.191	75.282	247.553
60	42	205	19.219	19.092	243.821	676.766
60	42	467	8.366	9.395	125.338	304.658
80	8	311	10.140	8.340	71.033	184.030
80	8	798	4.127	4.892	35.870	99.123
80	32	311	16.268	15.314	399.121	1001.748
80	32	798	6.196	6.789	181.964	527.012
80	56	311	12.478	14.837	548.807	1595.676
80	56	798	6.617	7.425	263.995	645.636
100	10	483	12.414	11.068	192.494	505.227
100	10	1245	3.854	5.300	86.643	280.434
100	40	483	13.844	14.995	561.683	1908.834
100	40	1245	5.688	5.688	255.230	637.596
100	70	483	12.052	12.314	936.565	3103.468
100	70	1245	7.295	7.295	512.813	1068.108
Mean			13.591	14.643	167.897	466.773

Bibliography

[1] Niels Agatz, Paul Bouman, and Marie Schmidt. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, 52(4):965–981, 2018.

[2] Antonio L. Alfeo, Mario G.C.A. Cimino, and Gigliola Vaglini. Enhancing biologically inspired swarm behavior: Metaheuristics to foster the optimization of uavs coordination in target search. *Computers & Operations Research*, 110:34 – 47, 2019.

[3] Jean-Philippe Aurambout, Konstantinos Gkoumas, and Biagio Ciuffo. Last mile delivery by drones: an estimation of viable market potential and access to citizens across european cities. *European Transport Research Review*, 11(1):30, Jun 2019.

[4] Gustavo S. C. Avellar, Guilherme A. S. Pereira, Luciano C. A. Pimenta, and Paulo Iscold. Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11):27783–27803, 2015.

[5] J. Bezos. Interview with Charlie Rose. broadcasted on: 60 minutes, cbs. December 1, 2013. (transcript).

[6] Maurizio Boccia, Adriano Masone, Antonio Sforza, and Claudio Sterle. Operational research methodologies for the pompeii archeological park. *Internal Report*, 2020.

[7] Paul Bouman, Niels Agatz, and Marie Schmidt. Instances for the tsp with drone, 2015. `http://dx.doi.org/10.5281/zenodo.22245`.

[8] Paul Bouman, Niels Agatz, and Marie Schmidt. Dynamic programming approaches for the traveling salesman problem with drone. *Networks*, 72(4):528–542, 2018.

[9] Nils Boysen, Dirk Briskorn, Stefan Fedtke, and Stefan Schwerdfeger. Drone delivery from trucks: Drone scheduling for given truck routes. *Networks*, 72(4):506–527, 2018.

[10] Nils Boysen, Stefan Schwerdfeger, and Felix Weidinger. Scheduling last-mile deliveries with truck-based autonomous robots. *European Journal of Operational Research*, 271(3):1085 – 1099, 2018.

[11] James Campbell, D.C. Sweeney, and Zhang J. Mobile Systems IV. Technical report.

[12] James F. Campbell, Ángel Corberán, Isaac Plana, and José M. Sanchis. Drone arc routing problems. *Networks*, 72(4):543–559, 2018.

[13] John Gunnar Carlsson and Siyuan Song. Coordinated logistics with a truck and a drone. *Management Science*, 64(9):4052–4069, 2018.

[14] N. Cary and N. Bose. Ups, fedex and amazon gather flight data to prove drone safety. venture beat. September 24, 2016.

[15] Bo Chen, Chris N. Potts, and Gerhard J. Woeginger. *A Review of Machine Scheduling: Complexity, Algorithms and Approximability*, pages 1493–1641. Springer US, Boston, MA, 1998.

[16] C. Cheng, Y. Adulyasak, and L.M. Rousseau. Formulations and exact algorithms for drone routing problem. *CIRRELT Report, 2018-31*, 2018.

[17] Cihan Tugrul Cicek, Hakan Gultekin, and Bulent Tavli. The location-allocation problem of drone base stations. *Computers & Operations Research*, 111:155 – 176, 2019.

[18] Roger Clarke. Understanding the drone epidemic. *Computer Law Security Review*, 30(3):230 – 246, 2014.

[19] Bruno N. Coelho, Vitor N. Coelho, Igor M. Coelho, Luiz S. Ochi, Roozbeh Haghnazar K., Demetrius Zuidema, Milton S.F. Lima, and Adilson R. da Costa. A multi-objective green uav routing problem. *Computers & Operations Research*, 88:306 – 315, 2017.

[20] Gloria Cerasela Crişan and Elena Nechita. On a cooperative truck-and-drone delivery system. *Procedia Computer Science*, 159:38 – 47, 2019. Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 23rd International Conference KES2019.

[21] Bart Custers. *Drones Here, There and Everywhere Introduction and Overview*, pages 3–20. T.M.C. Asser Press, The Hague, 2016.

[22] Bart Custers, JanJaap Oerlemans, and Bas Vergouw. The use of drones: An exploratory study on unmanned aerial vehicles (uavs). *The Hague, Boom Lemma Uitgevers.*, 2015.

[23] Konstantinos Dalamagkidis, Kimon P. Valavanis, and Les A. Piegl. *Aviation History and Unmanned Flight*, pages 11–42. Springer Netherlands, Dordrecht, 2012.

[24] Iman Dayarian, Martin Savelsbergh, and John-Paul Clarke. Same-Day Delivery with Heterogeneous Fleets of Drones and Vehicles. Technical report.

[25] Júlia Cária de Freitas and Puca Huachi Vaz Penna. A randomized variable neighborhood descent heuristic to solve the flying sidekick traveling salesman problem. *Electronic Notes in Discrete Mathematics*, 66:95 – 102, 2018. 5th International Conference on Variable Neighborhood Search.

[26] Mauro Dell'Amico, Roberto Montemanni, and Stefano Novellani. Drone-assisted deliveries: new formulations for the flying sidekick traveling salesman problem. *Optimization Letters*, 2019.

[27] Guilherme Dhein, Marcelo Serrano Zanetti, Olinto César Bassi de Araújo, and Ghendy Cardoso Jr. Minimizing dispersion in multiple drone routing. *Computers & Operations Research*, 109:28 – 42, 2019.

[28] Luigi Di Puglia Pugliese, Francesca Guerriero, Dimitrios Zorbas, and Tahiry Razafindralambo. Modelling the mobile target covering problem using flying drones. *Optimization Letters*, 10(5):1021–1052, Jun 2016.

[29] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, Jan 2017.

[30] Amro M. El-Adle, Ahmed Ghoniem, and Mohamed Haouari. Parcel delivery by vehicle and drone. *Journal of the Operational Research Society*, 0(0):1–19, 2019.

[31] Sergio Mourelo Ferrandez, Timothy Harbison, Troy Weber, Robert Sturges, and Robert Rich. Optimization of a truck-drone in tandem delivery network using k-means and genetic algorithm. *Journal of Industrial Engineering and Management (JIEM)*, 9(2):374–388, 2016.

[32] Quang Minh Ha, Yves Deville, Quang Dung Pham, and Minh Hoàng Hà. On the min-cost traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 86:597 – 621, 2018.

[33] Andy M. Ham. Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming. *Transportation Research Part C: Emerging Technologies*, 91:1 – 14, 2018.

[34] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126(1):106–130, 2000.

[35] Ho Young Jeong, Byung Duk Song, and Seokcheon Lee. Truck-drone hybrid delivery routing: Payload-energy dependency and no-fly zones. *International Journal of Production Economics*, 214:220 – 233, 2019.

[36] Aline Karak and Khaled Abdelghany. The hybrid vehicle-drone routing problem for pick-up and delivery services. *Transportation Research Part C: Emerging Technologies*, 102:427 – 449, 2019.

[37] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.

[38] Yanchao Liu. An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Computers & Operations Research*, 111:1 – 20, 2019.

[39] Raïssa G. Mbiadou Saleu, Laurent Deroussi, Dominique Feillet, Nathalie Grangeon, and Alain Quilliot. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks*, 72(4):459–474, 2018.

[40] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097 – 1100, 1997.

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

[42] Chase Murray and Ritwik Raj. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. Technical report, University of Buffalo, 2019. `https://papers.ssrn.com/abstract=3338436`.

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

[44] Charles Noon and James Bean. An efficient transformation of the generalized traveling salesman problem. *INFOR: Information Systems and Operational Research*, 31(1):39–44, 1993.

[45] Alena Otto, Niels Agatz, James Campbell, Bruce Golden, and Erwin Pesch. Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey. *Networks*, 72(4):411–458, 2018.

[46] Stefan Poikonen and Bruce Golden. The mothership and drone routing problem. *INFORMS Journal on Computing (to appear)*, 2019.

[47] Stefan Poikonen and Bruce Golden. Multi-visit drone routing problem. *Computers & Operations Research*, 113:104802, 2020.

[48] Stefan Poikonen, Xingyin Wang, and Bruce Golden. The vehicle routing problem with drones: Extended models and connections. *Networks*, 70(1):34–43, 2017.

[49] L. Di Puglia Pugliese and F. Guerriero. Last-mile deliveries by using drones and classical vehicles. *In: Sforza A., Sterle C. (eds) Optimization and Decision Science: Methodologies and Applications. ODS 2017. Springer Proceedings in Mathematics Statistics, vol 217. Springer*, pages 557–565, 2017.

[50] David Sacramento, David Pisinger, and Stefan Ropke. An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones. *Transportation Research Part C: Emerging Technologies*, 102:289 – 315, 2019.

[51] D. Schermer, M. Moeini, and O. Wendt. A variable neighborhood search algorithm for solving the vehicle routing problem with drones. *Technical Report BISOR-02/2018*, 2018.

[52] Daniel Schermer, Mahdi Moeini, and Oliver Wendt. Algorithms for solving the vehicle routing problem with drones. In Ngoc Thanh Nguyen, Duong Hung Hoang, Tzung-Pei Hong, Hoang Pham, and Bogdan Trawiński, editors, *Intelligent Information and Database Systems*, pages 352–361, Cham, 2018. Springer International Publishing.

[53] Daniel Schermer, Mahdi Moeini, and Oliver Wendt. A hybrid vns/tabu search algorithm for solving the vehicle routing problem with drones and en route operations. *Computers & Operations Research*, 109:134 – 158, 2019.

[54] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani. Unmanned aerial vehicles (uavs): A survey on civil applications and key research challenges. *IEEE Access*, 7:48572–48634, 2019.

[55] Seyed Mahdi Shavarani, Sam Mosallaeipour, Mahmoud Golabi, and Gökhan İzbirak. A congested capacitated multi-level fuzzy facility location problem: An efficient drone delivery system. *Computers & Operations Research*, 108:57 – 68, 2019.

[56] Reuters staff. Finnish post office tests drone for parcel delivery. September 14, 2015.

[57] Reuters staff. Russian postal drone program hits wall in debut. reuters. April 2, 2018.

[58] Marlin W. Ulmer and Sebastian Streng. Same-day delivery with pickup stations and autonomous vehicles. *Computers & Operations Research*, 108:1 – 19, 2019.

[59] Marlin W. Ulmer and Barrett W. Thomas. Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks*, 72(4):475–505, 2018.

[60] UPS. Ups tests residential delivery via drone. Feburary 2017. Note: YouTube Video.

[61] Bas Vergouw, Huub Nagel, Geert Bondt, and Bart Custers. *Drone Technology: Types, Payloads, Applications, Frequency Spectrum Issues and Future Developments*, pages 21–45. T.M.C. Asser Press, The Hague, 2016.

[62] Xingyin Wang, Stefan Poikonen, and Bruce Golden. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, 11(4):679–697, Apr 2017.

[63] Emine Es Yurek and H. Cenk Ozmutlu. A decomposition-based iterative optimization algorithm for traveling salesman problem with drone. *Transportation Research Part C: Emerging Technologies*, 91:249 – 262, 2018.

[64] Lu Zhen, Miao Li, Gilbert Laporte, and Wencheng Wang. A vehicle routing problem arising in unmanned aerial monitoring. *Computers & Operations Research*, 105:1 – 11, 2019.

[65] Dimitrios Zorbas, Luigi Di Puglia Pugliese, Tahiry Razafindralambo, and Francesca Guerriero. Optimal drone placement and cost-efficient target coverage. *Journal of Network and Computer Applications*, 75:16 – 31, 2016.