Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Computer Science

Bachelor Thesis

Wind field measurement in urban area using unmanned aircraft

Simona Musilová

Supervisor: Ing. Martin Selecký

Study programme: Open Informatics
Specialisation: Software systems

May 2017

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Computer Science

BACHELOR THESIS AGREEMENT

Student: Musilová Simona

Study programme: Open Informatics
Specialisation: Software systems

Title of Bachelor Thesis: Wind field measurement in urban area using unmanned aircraft

Guidelines:

1) Study the possibilities of wind measurements using unmanned aircraft.
2) Get familiar with representations of wind field and wind modeling.
3) Design an algorithm for trajectory planning for periodical wind measurements using unmanned aircraft in urban environment such that it minimizes the average information age of measured data. Take the wind field into account during the planning.
4) Implement the algorithm and verify it in a simulation using wind filed model and compare it with an algorithm that does not take wind field into account.

Bibliography/Sources:

[1] Ware, J. and Roy, N., 2016, May. An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In 2016 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1507-1514). IEEE.
[2] T. Bektas, The multiple traveling salesman problem: an overview of formulations and solution procedures, Omega, vol. 34, no. 3, pp. 209-219, 2006.
[3] G. Schrimpf, J. Schneider, H. Stamm-Wilbrandt, and G. Dueck, Record breaking optimization results using the ruin and recreate principle, Journal of Computational Physics, vol. 159, no. 2, pp. 139-171, 2000
[4] Garau, B., Alvarez, A. and Oliver, G., 2005, April. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach. In Proceedings of the 2005 IEEE international conference on robotics and automation (pp. 194-198). IEEE.

Bachelor Thesis Supervisor: Ing. Martin Selecký

Valid until the end of the summer semester of academic year 2017/2018

L.S.

prof. Dr. Michal Pěchouček, MSc.
Head of Department

prof. Ing. Pavel Ripka,CSc.
Dean

Prague, January 17, 2017

Declaration

I declare that I elaborated this thesis on my own and that I mentioned all the information sources and literature that have been used in accordance with the Guideline for adhering to ethical principles in the course of elaborating an academic final thesis.

In Prague on May 26, 2017 .....

Acknowledgements

I am especially grateful to my supervisor Ing. Martin Selecký for his valuable advice, patience and support during the creation of this thesis.

Furthermore, I would like to thank my partner Jan for his never-ending support throughout my studies. Also, I would like to thank my friend Hanka for her selfless help with the bachelor thesis proofreading and editing.

Finally, I thank my family and friends for support.

Abstract

The goal of this work is to design and implement a method for periodical wind measurements in urban environment using UAVs (Unmanned Aerial Vehicles) for improving CFD models. The wind field in given environment is taken into account during trajectory planning. The aim is to plan the path through a set of measuring locations such that the average information age of the measurements is minimized and limited UAV flight times are respected. This method is tested in simulated scenario with wind field modelled in CFD software. Planned paths are compared to paths planned by simple path planner which does not take the wind field into account.

Key words

wind field measurement, Unmanned Aerial Vehicle, path planning, geostatistical analysis, Monte Carlo Simulation, orienteering problem

Abstrakt

Cílem této práce je navrhnout a použít postupy pro pravidelná měření větru v městském prostředí za použití bezpilotních prostředů pro zpřesnění CFD modelů. Během plánování trasy jsou brána do úvahy větrná pole v daném prostředí. Záměrem je naplánovat trasu skrz množinu měřících bodů tak, aby byly shromážděné informace co nejaktuálnější při zachování co nejnižsího počtu přeletů bezpilotního prostředku. Tato metoda je testována na modelovém scénáři v prostředí s větrnými poli simulovanými pomocí CFD software. Naplánované cesty jsou porovnávány s cestami navrženými jednoduchým plánovačem tras, který nebere do úvahy větrná pole.

Klíčová slova

měření větrných polí, bezpilotní prostředek, plánování tras, geostatistická analýza, Monte Carlo simulace, orienteering problem

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Determination of wind field in urban environment is a challenging problem. Standard weather forecast models provide only large-scaled information about wind flow. Wind flow around buildings depends on building's size, shape, orientation and on the interaction of the building with its surrounding buildings [3]. Wind flow around buildings in a complex urban environment can change in its direction or velocity in a short time. For creating a model of the wind field in urban environment, new high-resolution methods need to be used.

There is a possibility of creating a model of the urban environment and simulate wind flow in it. Many different software provide the wind tunnel simulation usually based on the computational fluid dynamics solver. The advantage of this approach is the possibility of quick and simple change in resolution. On the other hand, the wind field models are very sensitive to initial conditions. By measuring the real wind field in environment and updating the wind field data, more precise models can be created.

One of the possibilities of wind field measurement in real urban environment is using unmanned aerial vehicles (UAVs). UAVs flying in urban environment need to be small enough to fly in narrow areas between buildings but robust enough to complete the journey. Additionally, UAVs need to be capable of carrying all measuring sensors.

The goal of this thesis is to design and implement an algorithm for optimal measurement of wind field in urban environment by one or more UAVs. The algorithm will take into account the wind field and maximum flight time of UAVs. The UAV traveling the planned path will visit defined set of measuring points in such an order, that it minimizes the average weighted information age of measured data. The set of measuring points is a subset of all given points which covers the whole area with minimal kriging error (see section 3.4.2). The task is then formulated as an orienteering problem, where the UAV (or team of UAVs) should visit these measuring points with respect to their limited flight time. The score of each measuring point, later used in orienteering problem (see section 3.5), is computed as variance of wind velocity at each point to ensure that the rate of visit of point where the wind velocity has been changing more is higher than the rate of visit of other points.

This thesis has been inspired by The Small Business Innovation Research (SBIR) program 'Microcosm Forecasting Utilizing Swarm Unmanned Aerial Vehicle Technology' [21] which was partitioned to three phases. The task of the first phase was to provide a review of available micro-UAVs and weather sensors. In the second phase the task was to develop a data gathering and control algorithms which should be used in the third phase in live and simulated testing. From gathered data the Three-Dimensional Wind Field (3DWF) wind model of urban environment was to be created. The 3DWF has been developed by the U.S. Air Force in conjunction with the U.S. Army Research Laboratory [7]. This thesis focuses on the phase two of the SBIR program.

This thesis is organized as follows: In chapter 2 related works analysing problems solved in this thesis are reviewed. Chapter 3 describes the problem into details and explains each part of it. In chapter 4 results of the methods used in one particular simulated scenario are provided. Lastly, chapter 5 concludes this thesis and presents some future remarks.

Chapter 2

Related work

This chapter provides the review of related works to this thesis. In the first section is the review of Unmanned Aerial Vehicles used for wind measurement and some of the sensors used to measure wind velocity. In the next section different approaches to path planning in different vector fields are provided. The next section provides review of works investigating the optimal placement strategy in different environments. In the last section of this chapter, the orienteering problem related works are provided with different solutions.

2.1 Hardware for wind measurement

There are many possible ways how to measure wind velocity. Different static sensors (e.g. meteorological observation towers [27]) are often used, but for the wind measurement in the urban environment they are not usable. Because of their size the static weather sensors cannot be build up in complex urban environment. The best way is to use Unmanned Aerial Vehicles (UAVs) with different sensors to fly in the urban area and measure all needed values. The advantage of using the UAV is the possibility of flying in different environments with the same UAV.

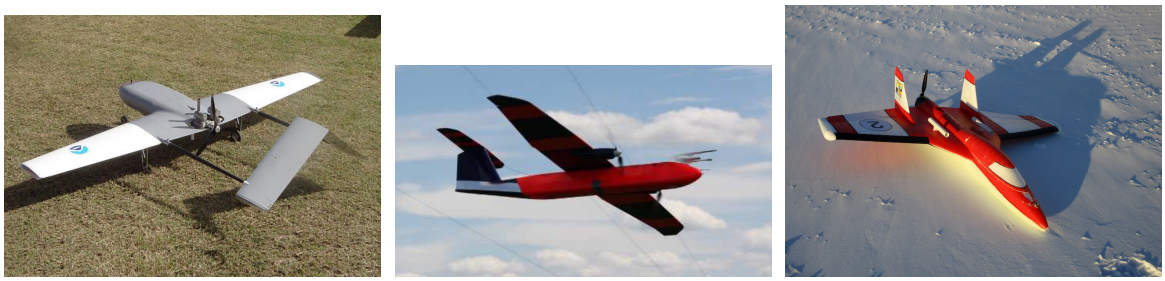
2.1.1 Unmanned Aerial Vehicles

Hardware possibilities for wind measurement were discussed in many surveys. Elston et al. [8] provide detailed overview of different UAVs and categorize UAVs to three groups according to their weight.

In the first category are UAVs with weight more than 10 kg but less than 30 kg. These UAVs provides the greatest payload capacity and endurance for the price of greater physical size and cost. One of the representatives of the Category I is Manta UAV designed and build by BAE Systems for the U.S. National Oceanic and Atmospheric Administration (NOAA) (see Figure 2.1a).

The Category II provides UAVs with smaller physical size and lower cost than Category I but the endurance and payload of Category II UAVs are lower. The weight of UAV is more than $1\,kg$ but less than $10\,kg$. The representative of this category is the Meteorological Mini Unmanned Aerial Vehicle (M^2AV) developed at the University of Braunschweig, Germany (see Figure 2.1b) [22].

In the last category are UAVs with weight less than $1\,kg$. Their payload and endurance are limited, which reduces their ability to carry multiple sensors [8]. On the other hand, their low cost and size are great for users with limited or no flight-related experiences. For example the Small Unmanned Meteorological Observer (SUMO) developed by the University of Bergen, Norway and Martin Müller Engineering (see Figure 2.1c).



(a) Category I Manta UAV (b) Category II M^2AV UAV (c) Category III SUMO UAV

Figure 2.1: Representative UAV of each category

2.1.2 Sensors for wind measurement

Moyano [19] in his master thesis provides an overview of the most common sensors and systems used to measure wind field characteristics. These sensors and systems can be used on board of UAVs.

A very simple sensor which measures only the wind speed and only in the horizontal plane is a cup anemometer (see Figure 2.2). This type of anemometer usually consists of 3 or 4 cups, each attached to a horizontal arm that is mounted on a vertical shaft. The wind speed estimation is based on the counted number of turns of the shaft. The cup anemometer cannot estimate the wind direction.

To measure both, the wind speed and the wind direction, a propeller anemometer, also known as a vane anemometer or a windmill anemometer, can be used (see Figure 2.3). The propeller anemometer consist of a vane and a propeller and provides only the horizontal measurement.

One of the most accurate sensors is a sonic anemometer which provides measurements of the wind speed and the wind direction in one-, two- or three-dimensional wind flow. The sonic anemometer consists of two transductors (see Figure 2.4). The measurement is based

Figure 2.2: Cup anemometer

Figure 2.3: Propeller anemometer

on the time that takes the signal to get from the transmitter to the receiver.

The other accurate sensor is a hot-wire anemometer which can be used to measure the wind speed. This type of anemometer consists of a fine wire which is heated by electric power (see Figure 2.5). As the air flows by, the wire cools down and the electrical resistance of used metal changes. The wind speed can be estimated based on the resistance of the wire.

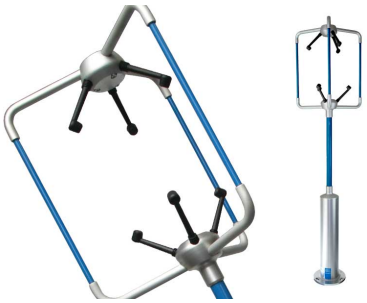


Figure 2.4: Sonic anemometer

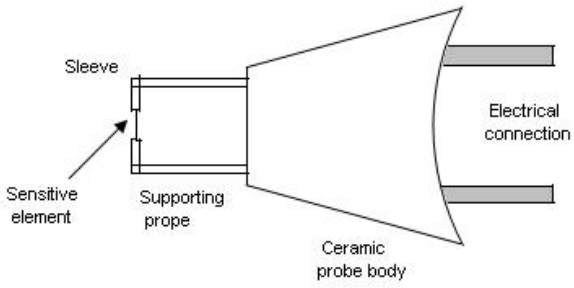


Figure 2.5: Hot-wire anemometer

To measure the wind speed and the wind direction a sphere anemometer can be used (see Figure 2.6). This anemometer consists of a sphere mounted on top of a rod which is used to collect the drag force. The resolution of the sphere anemometer is based on the frequency of the rod.

Next mentioned anemometer is a laser-cantilever anemometer which uses a laser and a position detector (see Figure 2.7). The detector is used to determine the deflection of the tip of a cantilever which is caused by the drag force. The laser-cantilever anemometer cannot be used on board of small UAVs because of its weight.

The very common sensor to indicate and measure airspeed of an aircraft is the Pitot tube (see Figure 2.8). The Pitot tube provides the possibility of measurement of the stagnation pressure and the static pressure. The aircraft velocity can be computed from Bernoulli's equation using the measured data. The stagnation pressure is measured at the very tip of the Pitot tube while the static pressure is measured from ports located on the airframe or

Figure 2.6: Sphere anemometer

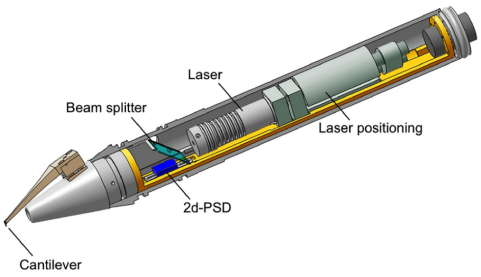


Figure 2.7: Laser-cantilever anemometer

from the ports on the side of the Pitot tube [8].

Another way of airspeed measurement is to use a multihole probe which provides the stagnation and static pressure measurements similar to the Pitot tube. The multihole probe contains multiple holes at different angles (see Figure 2.9) [8].

Figure 2.8: Pitot tube

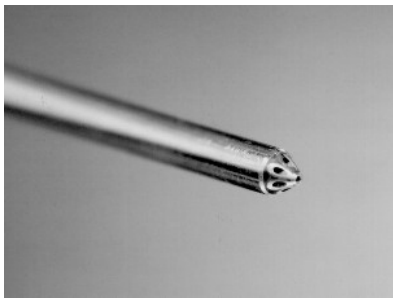


Figure 2.9: Multihole probe

The remote systems are based on the Doppler shift of emitted waves. The Radar measures the shift of electromagnetic waves reflected by areas with different density. The Sodar works on the similar principle but it emits sound waves. The Lidar's emitted electromagnetic waves are reflected by aerosol particles.

2.2 Planning in vector field

Different authors have been working on path planning in different vectors fields. Garau et al. [10] focused on the application of path planning algorithm in water current field. They used A* based algorithm to find optimum path of autonomous underwater vehicles (AUVs) in a given realistic ocean environment. As optimum path was considered path with minimum travel time. Authors' results show, that path planning is beneficial when the current speed is comparable to or higher than speed of the AUV.

Ware and Roy [25] used path planning algorithm in urban environment with respect to static wind field. They used 3D model of MIT campus with computational fluid dynamics solver to compute detailed wind field model. Authors validated the computed data from CFD solver by series of measurements in campus. A* algorithm was used to find minimum-energy trajectories, where energy was given as sum of kinetic and potential energy of UAV and the capacity of on-board battery. Ware and Roy compared the A* algorithm with naive algorithm and found out that the A* algorithm can significantly outperform the naive algorithm. Unlike naive algorithm, the A* planner avoids headwinds and uses tailwinds.

2.3 Optimal sample placement strategy

The goal of the optimal placement algorithm is to select a set of points from the given set of points which represents the phenomenon in given area the best. To select the best set of points the geostatistical analysis which assumes that there is a connection between the measured data value at a point and the location of the point in a given space is used.

According to Ballari et al. [2] the information about the phenomenon is given by a set of discrete observations which are interpolated to predict the probability of the observed phenomenon at unobserved locations. Next, the error is computed as the difference between the predicted data and the real data. The optimal placement is the one with the minimum total error.

Castello et al. [4] focused on optimal sensor placement in Wireless Sensor Network. In the first step, authors generated the grid to represent the given environment. Then the N random points were selected from the grid and variography was used to determine spatial dependency. Next, the ordinary point kriging was used to interpolate for predicted values between observed points. The mean variation of the kriging is computed for each repetition. The best solution is the one with the minimal mean variation. This process is called Monte Carlo Simulation. Authors' results show, that the minimum mean variation decreases as the number of the Monte Carlo repetitions increases.

Ordinary kriging was used to predict coverage in Wireless Local Area Network by Konak in [15]. Konak used ordinary kriging as a new tool to determine the optimal weights which produce the minimum path loss. The exponential function was used as the semivariogram function of distance between two points in space. The unknown values were estimated as weight-linear combination of known values from the given dataset.

2.4 Orienteering problem

Orienteering problem can be formulated as follows: In the given dataset of nodes in Euclidean plane find a path with maximum score between a starting node and a goal node. Each node is valued with a score $s_i \geq 0$, where the score of the starting node and the goal

node are $s_1 = s_n = 0$. The length (or duration) of the path is not greater than T_{max} [11]. The orienteering problem can be referred to as a Generalized Travelling Salesman Problem [23] where the salesman does not have to visit all cities but he knows the possible number of sales in each city. The goal of the salesman is to get the maximum number of sales within the maximum distance he can travel.

There are two main approaches to solve the orienteering problem. In the first approach the exact solution algorithm is used. Laporte and Martello [16] described the problem as an integer linear programming problem and used a constraint relaxation to the model. Then the problem was solved through linear programming and additional conditions were introduced through a branch and bound process. Leifer and Rosenwein [17] added some valid constraints to the previous method and introduced more effective cutting plane method. Fischetti et al. [9] created branch-and-cut algorithm, which can plan a path in the set with up to 500 nodes.

The second approach to solve the orienteering problem are heuristic methods which can be more efficient than the exact solution. Tsiligirides [23] provides the stochastic algorithm (S-algorithm) to solve the orienteering problem which uses the Monte Carlo Technique. This algorithm is based on generating many possible paths and selecting the best one. Tsiligirides also provides the deterministic algorithm (D-algorithm) which is based on Wren's and Holliday's work [26]. In this algorithm the path is build in several sectors according to given rules.

Golden et al. [11] provides a centre of gravity heuristic algorithm which works in four steps. In the first step, the relatively high score path from the starting node to the goal node is created. In the next step the interchange procedure is used to find shorter path followed by the insertion where as many nodes as possible are inserted into the path without violating T_{max}. In the last step all nodes are ranked based on their distance from the centre of gravity from the previous path, starting with the empty path. The cheaper the node is, the sooner is inserted to the path.

Chao et al. [5] provides the five-phase heuristic method, which consists of initialization step and improvement step. In the initialization step the path is created in the selected ellipse where only the insertion cost is taken into account, the score of each node is ignored. Within the ellipse, more paths can be created but each of them has to fulfil the T_{max} constraint. From all paths, the one with the highest score is selected. In the next step the solution is improved by two-point exchange. Point from selected path and point which is not in selected path are exchanged and the new score of all available paths is recomputed and the best path is selected.

Definitions of other orienteering problems were discussed in [24] where Vansteenwegen et al. provide problem definitions and mathematical formulations of the orienteering problem, the team orienteering problem, the orienteering problem with time windows and the team orienteering problem with time windows, each kind of orienteering problem with solution approaches and practical applications.

Chapter 3

Methodology

In this chapter the created algorithm is described. In the first section the algorithm is briefly introduced. In the next section the creation of the model of the environment and the simulation of wind field is described. Section 3.3 provides detailed information about used A* algorithms. In section 3.4 the model and the A* algorithm are used in the geostatistical analysis. Lastly, section 3.5 provides the solution to the orienteering problem.

3.1 Solution design

The input data for this method are the start point and the goal node of path and the map of the urban environment. In the first step, the model of urban environment was created from the provided map. This model was imported into computational fluid dynamics solver to simulate the wind flow and to get the input data for the algorithm.

Next, the path planner based on A* algorithm was created. In this thesis, two versions of the A* algorithm are used, a simple A* planner and a wind A* planner. The simple A* planner is used to determine the shortest path between two nodes in environment with obstacles used in the geostatistical analysis. The wind A* planner is used to plan the shortest path and compute the travel time between two nodes in the given environment and to compute the matrix for the orienteering problem.

In the next step the representative grid of nodes was selected. This grid equally covers the whole area avoiding all objects. The geostatistical analysis is based on the Monte Carlo Simulation which runs a given number of times and from the provided representative grid of nodes selects a set of measurement locations with the minimum wind estimation error.

Next, the orienteering problem is iteratively used to plan a path between the given set of nodes in a given time budget in order to maximize the collected score (see section 3.5). The score of each point is based on variance of the wind velocity in different time stamp from the CFD simulation. The more the wind velocity has been changing in a node, the higher the score of this node is thus ensuring that the locations with the more variable wind velocity

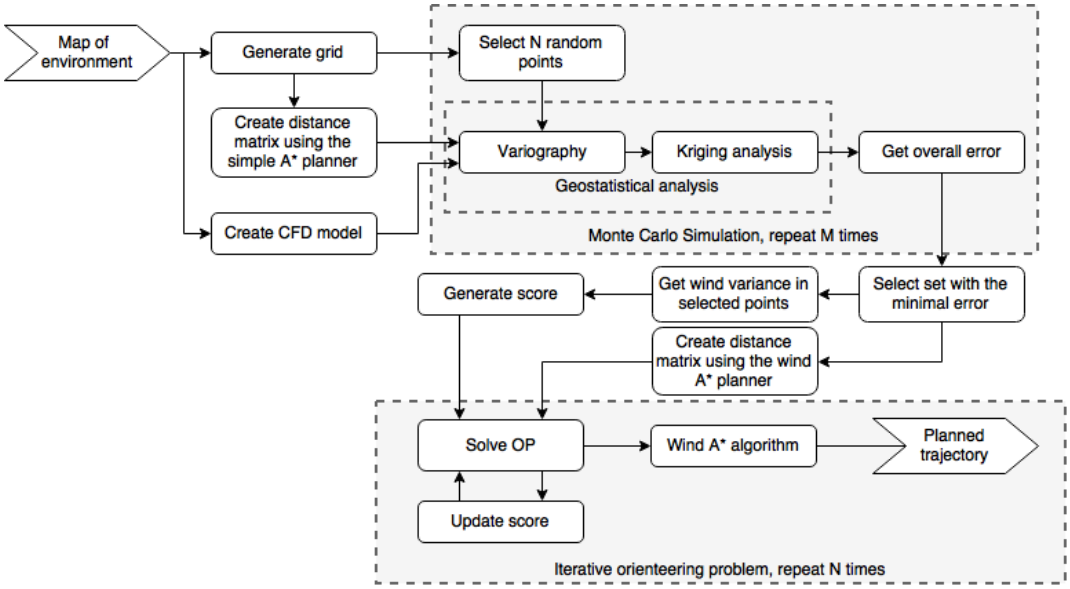


Figure 3.1: Algorithm procedure

are visited more often. The solution of the orienteering problem is the best planned path in the given environment.

The orienteering problem mentioned in this thesis plans path iteratively to minimize the average information age of measured data. If the number of used UAVs is higher, the team orienteering problem should be used. The goal of the team orienteering problem is to plan a given number of paths, each limited by the time budget, that maximises the total collected score [24]. In our case we assume that all the UAVs would have common base and so each path would start in the same starting node and ends in the same goal node. Another possible method to solve the iterative orienteering problem is utilizing the same orienteering problem solver but updating the score and information age of visited locations after the paths for all of the UAVs are found. To prevent multiple visits of measuring locations by different UAVs during one iteration, the scores of locations already visited by other UAVs in particular iteration are temporarily set to zero.

3.2 Model of environment

The Autodesk AutoCAD for Mac 2017 was used to create the model of city and the Autodesk CFD 2017 software was used to simulate the wind flow. The CFD software was used because it provides all the information needed in this thesis and thanks to student's licences from Autodesk the usage of both software was free.

As a modelling software the Autodesk CFD software [12] was used. The CFD is a simulation software which uses the computational fluid dynamics to compute and visualize the flow

around or inside objects. The model of the urban environment was created in the AutoCAD, exported to a file and later imported into the CFD software.

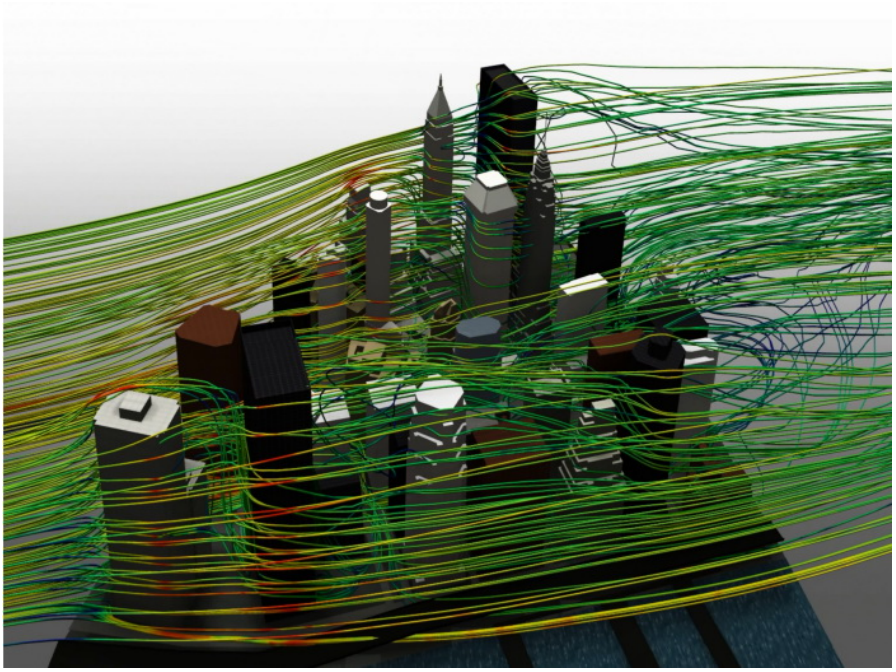


Figure 3.2: Visualization of wind field in urban environment using CFD[1]

In this thesis we assume only one main direction of wind, from south to north. If the main wind direction in provided environment changes, the offline simulation needs to be done for each direction. Then according to the real wind direction the correct dataset is chosen.

The CFD software provides possibilities of computing different values but in this thesis only the wind velocities were computed. As a solution mode was used a transient mode which provides the time-dependent analysis. To run the transient analysis following parameters are necessary [13]:

- **Time Step Size:** The Time Step Size is always given in seconds.

- **Stop Time:** The simulation can be stopped when the given stop time is reached or when the given number of steps has passed or whichever comes first.

- **Inner Iterations:** The number of inner iterations for each time step during analysis. For each time step the iterations need to be run since the CFD uses the implicit method to discretize the flow equations and solves the equations at each inner iteration.

[1] `http://isicad.ru/ru/articles.php?article_num=16431`

The path planning algorithm in this thesis works with representative static wind field which is given by wind vectors in each node from the previous CFD simulation. All input data need to be in text document with following data format on each line:

$$x\ y\ w_x\ w_y$$

where (x, y) is position of node in given environment and (w_x, w_y) is the wind vector. The document contains one line of text for each node. The more data from different time stamps is provided, the more optimal path is planned by algorithm.

Even though the UAVs are capable of changing their airspeed, the speed of UAV is considered to be constant during the whole time. We assume that the forward airspeed is not changing while turning.

3.3 A* algorithm

We assume that the UAV is not able to change its altitude which may be restricted by legislation or the air traffic control. In that case, UAV cannot fly over a building so the path planning algorithm will plan the path on the two-dimensional plane avoiding buildings.

As a path planning algorithm, the A* algorithm is used. The A* algorithm is an informed search algorithm, which searches for the path with the lowest cost between the start node and the goal node. The total cost of each node in a graph is given as follows:

$$f(n) = g(n) + h(n),$$

where n is the last node of the path, $g(n)$ is the total cost of the path from the start node to node n and $h(n)$ is an estimated cost of the path from node n to the end node, called heuristic function. Additionally, the heuristic function must satisfy following condition:

$$\forall(n) : 0 \leq h(n) \leq h^*(n), \tag{3.1}$$

where $h^*(n)$ is the real cost of the path from node n to the goal node. In such a case, the A* algorithm will find the cheapest path.[20]

In this thesis the minimum travel time between two nodes is considered as the cost of the shortest path. The evaluation of node n is given as follows:

$$t(n) = t_h(n) + t_g(n),$$

where $t_h(n)$ is the estimation of travel time from node n to the goal node and $t_g(n)$ is the travel time from the start node to node n.

This thesis works with two variations of the A* algorithm for path planning. First is a simple A* planner which does not take the wind field into account. This version is used in section 3.4.1 to determine the travel time and the distances between two nodes for semivariogram estimation. The second A* planner is a wind A* planner with knowledge about the wind field which plans the path based on given data. This version is used to determine the shortest travel time between two points and its travel time in the wind field. It is used to compute the travel time matrix which is used in the orienteering problem (see section 3.5). Each of those variations will use different equations to calculate the cost of path.

3.3.1 Simple A* planner

The simple A* planner computes the heuristic travel time as follows:

$$t_h(n) = \frac{s_E(n)}{v},$$

where $s_E(n)$ is an Euclidean distance between node n and the goal node, v is the speed of the UAV when the wind speed is zero. For the total cost of the path from the start node to node n, the simple A* planner uses similar equation:

$$t_g(n) = \frac{s(n)}{v},$$

where $s(n)$ is the travelled distance between the start node and node n.

3.3.2 Wind A* planner

The heuristic travel time of the second A* planner is computed as follows:

$$t_h(n) = \frac{s_E(n)}{v + w_{max}},$$

where w_{max} is the maximum wind speed in the given data set. The heuristic cost given by this simple function is always lower then the actual cost of path from the given node n to the goal node [10], so it fulfils the A* condition (3.1).

To compute the most real cost of the path between nodes A and B, the path is divided in the middle by point S and the total travel time is then given by sum of the travel times in each section. In the first section wind velocity $\vec{w}_A$ in node A is taken into account in equations while in the second section wind velocity $\vec{w}_B$ in node B is taken into account. In the following text only equations in the first section will be introduced since equations in the second section are similar.

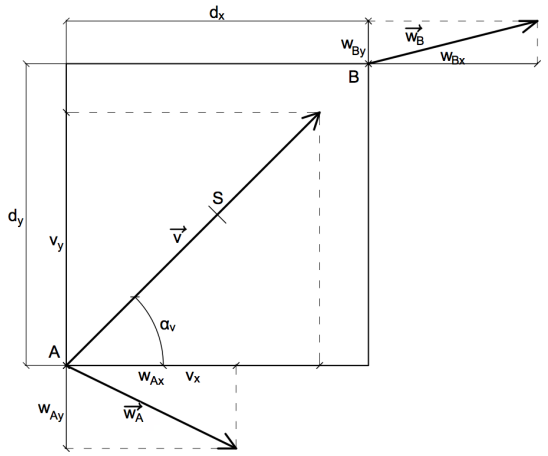


Figure 3.3: Visualization of wind velocity vectors ($\vec{w_A}$, $\vec{w_B}$) and UAV velocity vector ($\vec{v}$)

According to Figure 3.3 the distance in the first section between node A and point S is given as follows:

$$\vec{d_A} = t_A(\vec{v} + \vec{w_A}),$$

where t_A is travel time between node A and point S, $\vec{v}$ is the velocity of UAV and $\vec{w_A}$ is wind velocity in node A. Travel distances on x and y axis are given as follows:

$$d_{Ax} = vt_A \cos\alpha_v + w_{Ax}t_A, \tag{3.2}$$
$$d_{Ay} = vt_A \sin\alpha_v + w_{Ay}t_A, \tag{3.3}$$

where $\vec{w_A} = (w_{Ax}, w_{Ay})$ is the wind velocity vector. Using equations (3.2) and (3.3) with the fact that $\sin^2 x + \cos^2 x = 1$, only one equation is created, which after some editing is as follows:

$$t_A^2(w_{Ax}^2 + w_{Ay}^2 - v^2) + t_A(-2d_{Ax}w_{Ax} - 2d_{Ay}w_{Ay}) + d_x^2 + d_y^2 = 0,$$

This equation can be simplified by the fact, that point S is right in the middle of nodes A and B, which means, that $d_{Ax} = d_{Bx} = \frac{1}{2}d_x$ and $d_{Ay} = d_{By} = \frac{1}{2}d_y$.

The final equations to compute travel time between nodes A and B are as follows:

$$t_A^2(w_{Ax}^2 + w_{Ay}^2 - v^2) + t_A(-d_x w_{Ax} - d_y w_{Ay}) + (\frac{1}{2}d_x)^2 + (\frac{1}{2}d_y)^2 = 0,$$
$$t_B^2(w_{Bx}^2 + w_{By}^2 - v^2) + t_B(-d_x w_{Bx} - d_y w_{By}) + (\frac{1}{2}d_x)^2 + (\frac{1}{2}d_y)^2 = 0,$$
$$t_g = t_A + t_B.$$

3.4 Geostatistical analysis

The geostatistical analysis is based on theory, that the measured data values depend on the location of the measuring point. Using the multiple repetition of the Monte Carlo Simulation provides the coverage of given area with small set of points with minimal reconstruction error when using ordinary point kriging [4].

In the first step the classical variography is used where a variogram describes the dependency between an observation within the analysed plane [4]. The variogram at unobserved points can be calculated through known values at observed points. This function is called the semivariogram function. In the next step of the geostatistical analysis, the ordinary point kriging is used to interpolate the observations to the grid. The ordinary point kriging uses the weighted averages of the measured data values in neighbouring points to predict the unmeasured values at unobserved locations. To get an estimation error the predicted

value is compared to the real value. The optimal placement of selected points is given by the minimal estimation error of all points.

3.4.1 Data preparation

Because of the given dataset of nodes is too large, the smaller representative set of nodes is selected. Those nodes equally cover the whole area avoiding all obstacles and create set $G(n)$.

For each pair of nodes from set $G(n)$ the travel time is computed using the simple A* planner, mentioned in Section 3.3.1. Since we assume the constant forward airspeed v of the UAV the shortest distance $h_{i,j}$ between nodes i and j can be estimated as travel time $t_{i,j}$. The distance can be computed as follows:

$$h_{i,j} = v\, t_{i,j}.$$

Each travel time is stored in matrix

$$M_{sim} = \begin{bmatrix} 0 & t_{1,2} & \dots & t_{1,N} \\ t_{2,1} & 0 & \dots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \dots & 0 \end{bmatrix},$$

where $t_{i,j}$ is travel time between node on position i in set $G(n)$ and node on position j in set $G(n)$. Because of simple A* algorithm is used, matrix M_{sim} je symmetric. For each path between nodes on position i and on position j the travel time is

$$t_{i,j} = t_{j,i}.$$

Matrix M_{sim} is saved to a file, so it can be loaded and reused later.

3.4.2 Monte Carlo Simulation

The first step of Monte Carlo Simulation is to randomly select N nodes from set $G(n)$. Selected nodes create set $V(n)$.

In the next step, the ordinary point kriging is used, which is the most commonly used type of kriging [15]. In this method, the unknown wind velocity $\hat{w_u}$ in node u is based on weighted averages and observations of nodes in set $V(n)$ as follows [4]:

$$\hat{w_u} = \sum_{i=0}^{N} \varphi_i w_i \tag{3.4}$$
$$where \sum_{i=0}^{N} \varphi_i = 1$$

where φ_i is the weight of node i and w_i is the known wind velocity at point i from the given dataset.

The kriging method is used to calculate weights which produce the minimal error. The weights are calculated as follows [15]:

$$\begin{bmatrix} \varphi_1 \\ \vdots \\ \varphi_N \\ \lambda \end{bmatrix} = \begin{bmatrix} \gamma(h_{1,1}) & \dots & \gamma(h_{1,N}) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(h_{N,1}) & \dots & \gamma(h_{N,N}) & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \gamma(h_{1,u}) \\ \vdots \\ \gamma(h_{N,u}) \\ 1 \end{bmatrix}$$

where $h_{i,j}$ is distance between nodes on position i and j in set $V(n)$, $\gamma(h_{i,j})$ is a semivariogram function of distance between nodes i and j, λ is the Lagrange multiplier to minimize the kriging error and u is the node with unknown wind velocity. The semivariogram function is given as exponential function as follows [1]:

$$\gamma(h_{i,j}) = \begin{cases} 0 & h_{i,j} = 0 \\ C_0 + (C_1 - C_0) exp(1 - 3h_{i,j}/R) & h_{i,j} > 0 \end{cases} \tag{3.5}$$

where C_0 is the nugget effect, which represents variability at distances smaller than the typical sample spacing in the dataset, C_1 is the still parameter, which is the maximum value of the semivariogram function, and R is the range parameter, which defines the distance beyond which the correlation between two points is assumed to be essentially zero.

The node u is selected from nodes which are in the set $G(n)$ but not in the set $V(n)$. All those node create set $P(n) = G(n) \setminus V(n)$. Wind velocities in those nodes can be predicted from the wind model estimated by the CFD solver. An error can be computed by comparing the wind velocity w_u given by the CFD model with the computed wind velocity from

equation (3.4).

The process described above is only for one node u. To get the overall error e of the sample placement, the error of wind estimation at each point needs to be combined as follows:

$$e = \sum_{u \in P(n)} \hat{w_u} - w_u, \tag{3.6}$$

This whole process runs a given number of times M. In each iteration, the overall error of wind estimation is computed and the set with the minimum error is chosen as the best set $B(n)$ of nodes to cover the given area.

3.5 Orienteering problem

The problem of finding the best order of the measuring nodes in the path that would minimize the weighted information age and respect the maximum flight time of the UAV can be modelled as the orienteering problem [24]. The orienteering problem's goal is to plan a path with the maximum collected score within the given time budget T_{max} from the start node to the goal node. To solve the orienteering problem a set of nodes $V(n)$ is given, each node i with its score $s(i)$. Travel time $t_{i,j}$ from each node on position i in set $V(n)$ to each node j in set $V(n)$ is also given. The score of each node is additive and each node can be visited only once [24].

In this thesis the iterative orienteering problem solution is provided. The algorithm runs a given number of iterations. In each iteration the path is planned with respect to the orienteering problem's goal mentioned above. The score of each node changes after every iteration. If the node was visited in the last planned path, its score resets to its original value. The score of each node which was not visited increases by the constant 5 which is the half of the maximum score. This simple algorithm was chosen as compromise to ensure that the score of each point increases equally while keeping the time needed to visit nodes with the lower initial score minimized.

For each node from set $V(n)$ its score is computed based on the wind velocity history in the node. The sample variance of wind velocity in x axis at each point is given as follows:

$$r_x^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\bar{w}_x - w_{xi})^2,$$

where N is number of history data in the point, w_{xi} is the wind velocity in time stamp i and $\bar{w}_x$ is the average wind velocity in x axis in all given history data. The variance for y axis is computed similarly and then it is summed together as follows:

$$r^2 = \sqrt{(r_x^2)^2 + (r_y^2)^2},$$

where the r_x^2 is the variance in x axis and r_y^2 is the variance in y axis.

The set of points was divided into 10 sections according to the normalized variance value of each point. The normalized value was computed as follows:

$$s_i = 100\frac{r_i}{r_{max}},$$

where s_i is the score at point i, r_i is the variance value at point i and r_{max} is the maximum variance value in the set. Points in each section are valued a score in range $1 - 10$. The first section, with normalized variance in range $\langle 0 - 10)$ is valued with score 1, next section, with normalized variance in range $\langle 10 - 20)$ is valued with score 2 and so on. The last section, with normalized variance in range $\langle 90 - 100\rangle$ is valued with score 10.

For each pair of nodes from set $V(n)$ the travel time is calculated and its value is stored in a matrix

$$M_{wind} = \begin{bmatrix} 0 & t_{1,2} & \dots & t_{1,N} \\ t_{2,1} & 0 & \dots & t_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N,1} & t_{N,2} & \dots & 0 \end{bmatrix},$$

where $t_{i,j}$ is travel time from node on position i in set $V(n)$ to node on position j in set $V(n)$. To compute each travel time the wind A* planner is used (see Section 3.3.2).

Matrix M_{wind} is not symmetrical since travel time from node on position i to node on position j in set $V(n)$ can be different than travel time in opposite direction. Created matrix M_{wind} is saved to a file, so it can be reused later.

The orienteering problem can be formulated as an integer problem [24] as follows:

$$max \sum_{i=2}^{N-1} \sum_{j=2}^{N} s_i x_{ij}, \tag{3.7}$$

$$\sum_{j=2}^{N} x_{1j} = \sum_{i=1}^{N-1} x_{iN} = 1, \tag{3.8}$$

$$\sum_{i=1}^{N-1} x_{ik} = \sum_{j=2}^{N} x_{kj} \leq 1; \forall k = 2, \ldots, N-1, \tag{3.9}$$

$$\sum_{i=1}^{N-1} \sum_{j=2}^{N} t_{ij} x_{ij} \leq T_{max}, \tag{3.10}$$

$$2 \leq u_i \leq N; \forall i = 2, \ldots, N, \tag{3.11}$$

$$u_i - u_j + 1 \leq (N-1)(1 - x_{ij}); \forall i, j = 2, \ldots, N, \tag{3.12}$$

$$x_{ij} \in 0, 1; \forall i, j = 1, \ldots, N, \tag{3.13}$$

where $x_{ij} = 1$ if visit of node i is followed by visit of node j, otherwise $x_{ij} = 0$ and u_i is the position on node i in the path. The objective function (3.7) is to maximise the total collected score. Constraint (3.8) guarantee that the path starts in node 1 and ends in node N. Constraints (3.9) ensure the connectivity of the path and guarantee that every node is visited at most once. Constraint (3.10) ensures the limited time budget. Constraints (3.11) and (3.12) are necessary to prevent subtours [18].

In this thesis, to solve the orienteering problem, the Variable Neighborhood Search (VNS) algorithm from the Java Metaheuristics Search framework (JAMES) was used[2].

[2] www.jamesframework.org

Chapter 4

Results

In the following section, model of one particular scenario is described. Later in this chapter results of simulations using this model are provided.

4.1 Configuration

The created model contains of 18 buildings, each building with the minimum height of $20\,m$. The size of whole model is $450 \times 482.5\,m$. The model represents the simplified city where wide long streets are present as well as narrow short ones. In the model a large free space is representing a square or a parking lot. On the map, see Figure 4.1, the size of each building can be seen. The height of each building is written in the top left corner of the building.

Around the model a wind tunnel was created to simulate the wind flow. The height of the wind tunnel was $150\,m$, the length of the tunnel in the perpendicular direction to the wind flow was $2000\,m$ and the length of the tunnel in the parallel direction to the wind flow was $3000\,m$. The initial wind speed in all simulation was

$$w_0 = 10\,m.s^{-1}$$

and the initial wind direction was from the south to the north or from the bottom to the top of each picture in this thesis.

The model of environment was placed in the middle of the wind tunnel in the x axis, to the bottom in the z axis to guarantee the air does not flow under the buildings. In the y axis the model was placed $900\,m$ from the beginning of the wind tunnel to create enough space for wind flow in front of and behind the model [14].

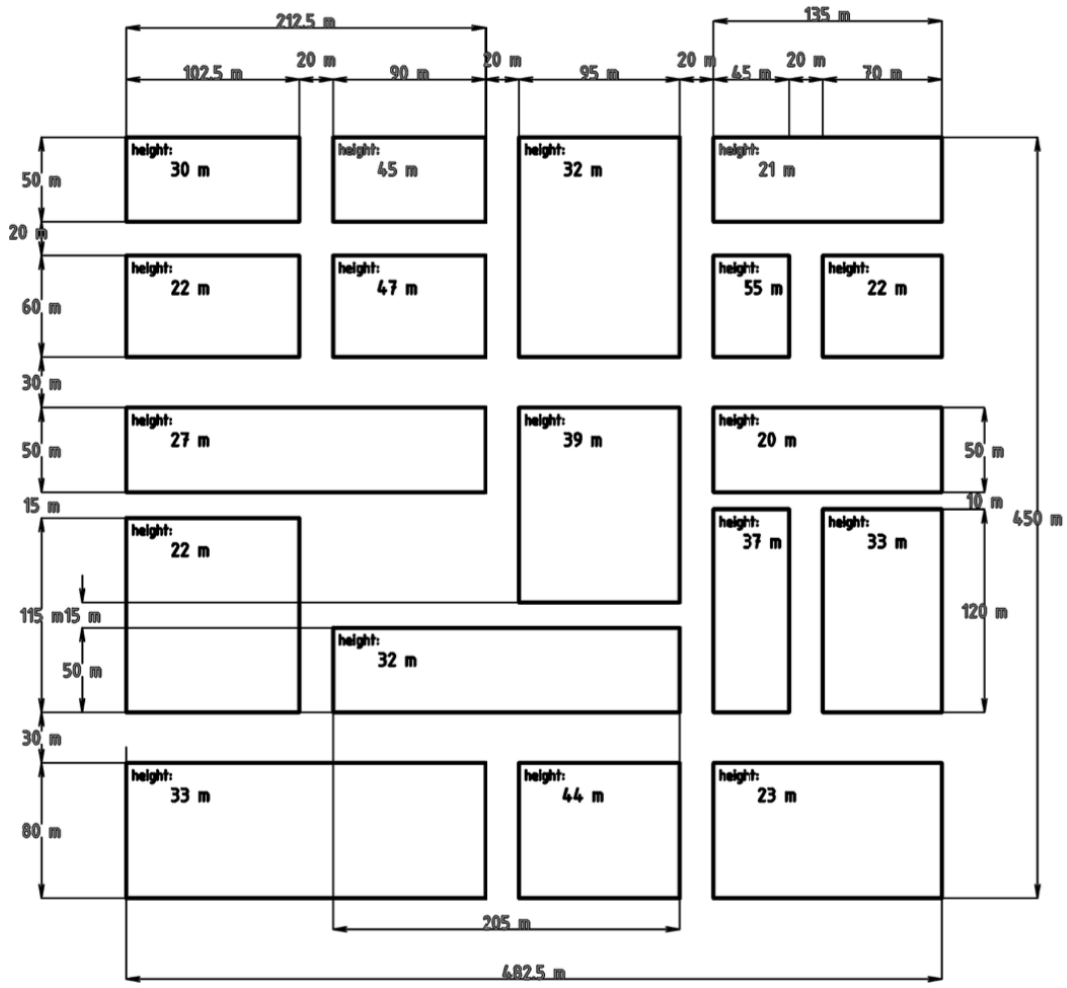


Figure 4.1: Model of urban environment from top-view

The simulation in wind tunnel was stopped several times to gather the measured data. The Table 4.1 shows the configuration of the running simulation in transient mode using the Autodesk CFD software. The time step size and the number of inner iterations is always the same. Only difference between each simulation is the stop time. Data from different time stamps are later used in the orienteering problem to get the score of each node (see Section 3.5).

Simulation	1	2	3	4	5	6
Stop time	1800 s	3600 s	5400 s	7200 s	9000 s	10800 s
Time Step Size	20 s					
Inner Iteration	1					

Table 4.1: Configuration of the simulations in transient mode in Autodesk CFD

The data set used later in planning is from time stamp $t = 3600$ s. In this time step, the

wind velocities are stable and path between each two nodes does exists.

In the next step, the measuring points were created. All measuring points are in the same height, in this thesis in the height of 20 m. Measured data was discretized over a grid where the distance between neighbouring points of the grid was 5 m. Measuring point was created at each point of the grid, where it was possible (free space without any objects). The set of measuring points covers the whole model and additional area of 15 m (3 rows of points) on each side of the model (see Figure 4.2). To automatize the process of creating the measuring points, the Python script was created.

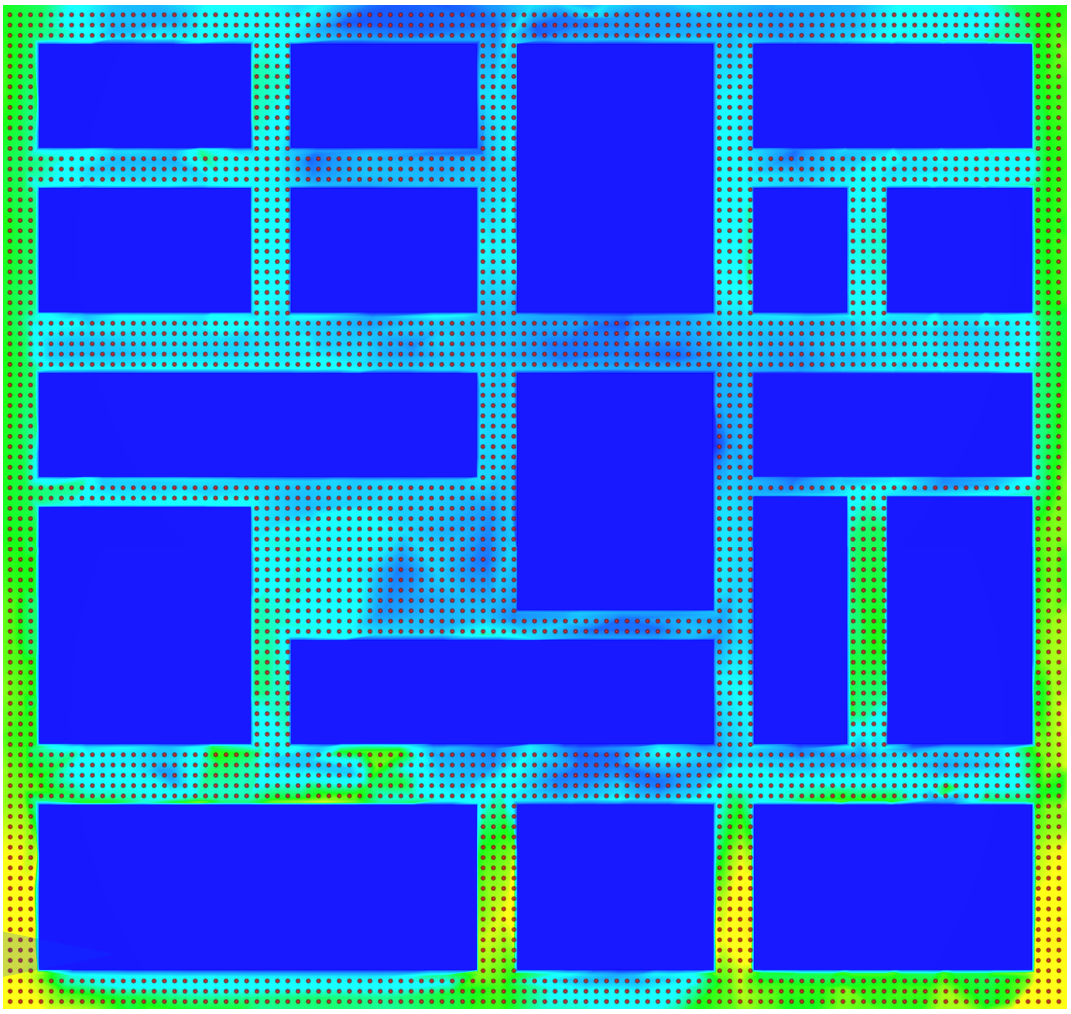


Figure 4.2: Position of measurind points

The speed of UAV is considered to be constant:

$$v = 10\, m.s^{-1}.$$

The given grid of all nodes was divided into 17×17 sections. This number of sections was chosen as a compromise between good are coverage and reasonable amount of points for computational feasability. From each section the most cantered node which is also in the given dataset was chosen. The number of all chosen nodes was 169 and they created set $G(n)$.

4.2 Points selection

The semivariogram function showing correlation of wind vector at given locations given their mutual distance can be seen on Figure 4.3. These data were taken from the CFD model and all needed values for equation (3.5) can be taken from it. The values of the radius and still parameters are shown on the picture and the nugget effect is zero. The selected values are as follows:

Parameter	Value
nugget	$C_0 = 0$
still	$C_1 = 9$
radius	$R = 450$

Table 4.2: The nugget, radius and still parameters

Number of sample nodes was chosen to be $N = 30$ as compromise between good area coverage and feasibility of their coverage with small number of UAVs. The Monte Carlo simulation ran multiple times with different number of iteration. Each simulation provided different nodes selection with different errors (see Figure 4.5). The overall estimation error e can by computed using the equation (3.6). The estimation error of each simulation run is shown in Figure 4.4. The higher the number of iterations in Monte Carlo simulation is, the better result is provided (result with the lower estimation error). Next the selected points with the lowest estimation error is used (see Figure 4.5f) The selected nodes create the set $B(n)$.

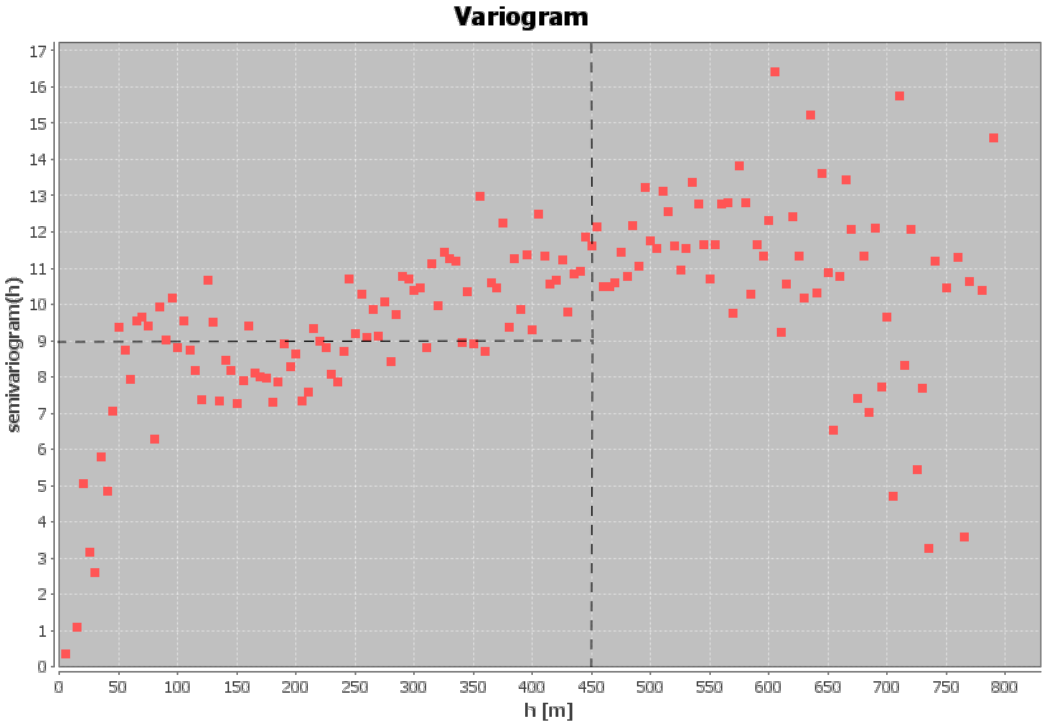


Figure 4.3: Variogram showing the correlation of points given their mutual distance

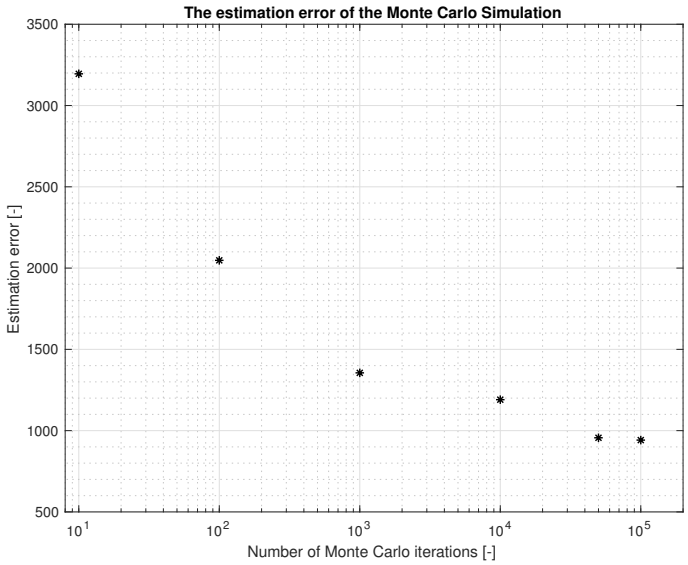
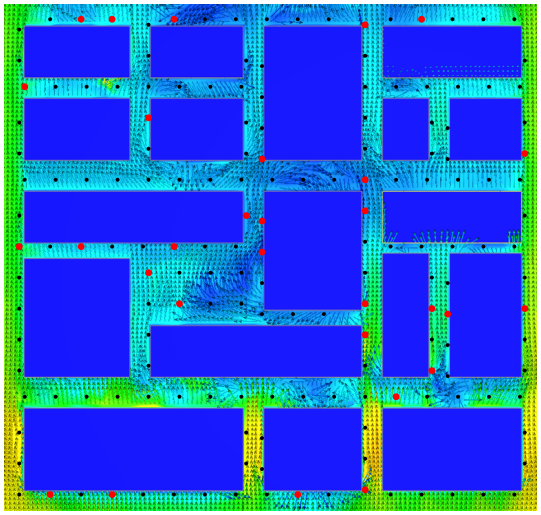
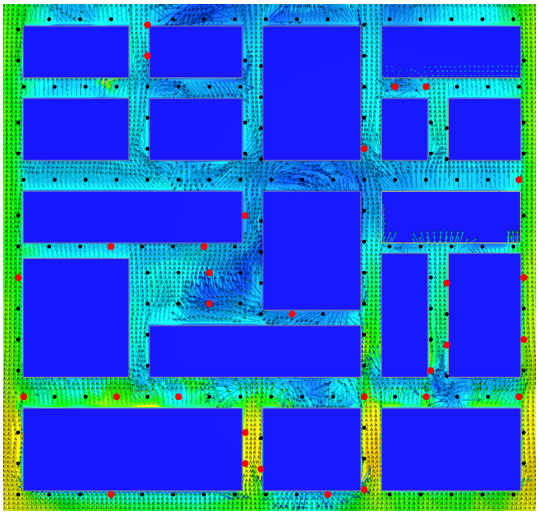


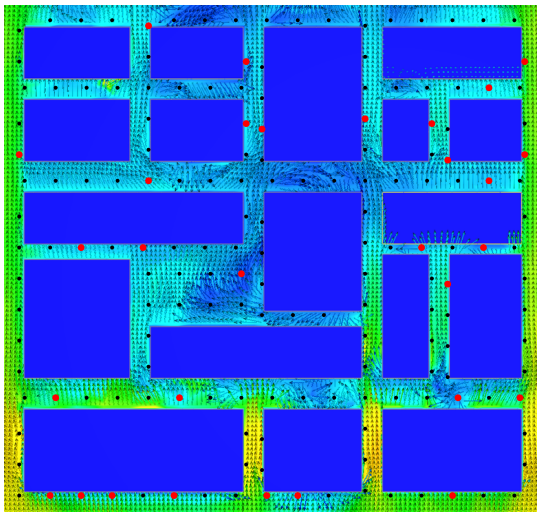
Figure 4.4: Dependence of the number of iteration in Monte Carlo simulation to estimation error



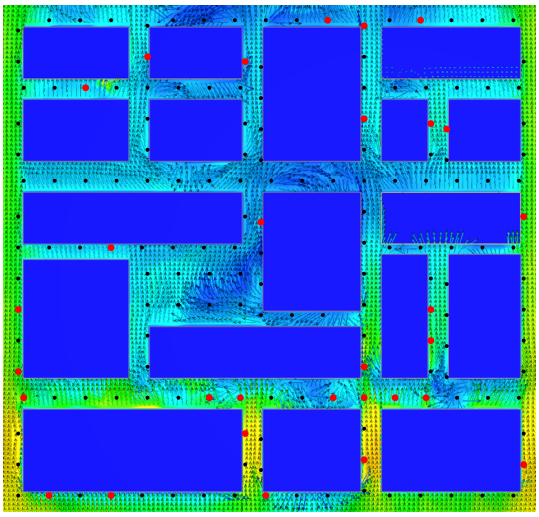
(a) $n_i = 10,\ e = 3195.05$



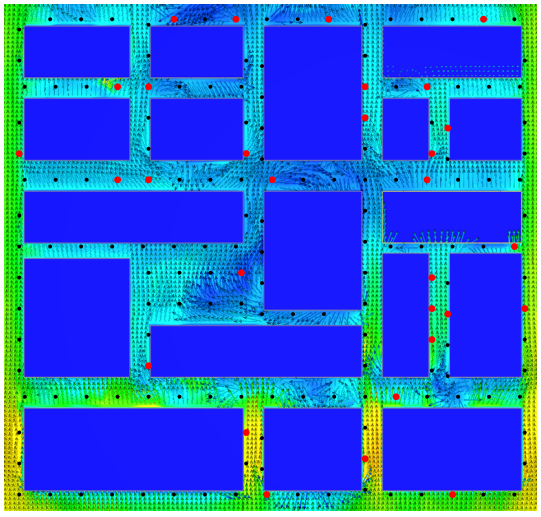
(b) $n_i = 100,\ e = 2047.64$



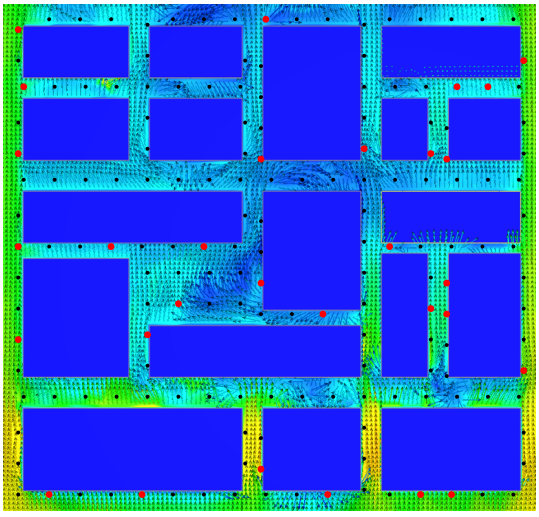
(c) $n_i = 1\,000,\ e = 1355.47$



(d) $n_i = 10\,000,\ e = 1190.52$



(e) $n_i = 50\,000,\ e = 956.05$



(f) $n_i = 100\,000,\ e = 941.78$

Figure 4.5: Results of Monte Carlo Simulation for given number of iterations n_i with the estimation error e.

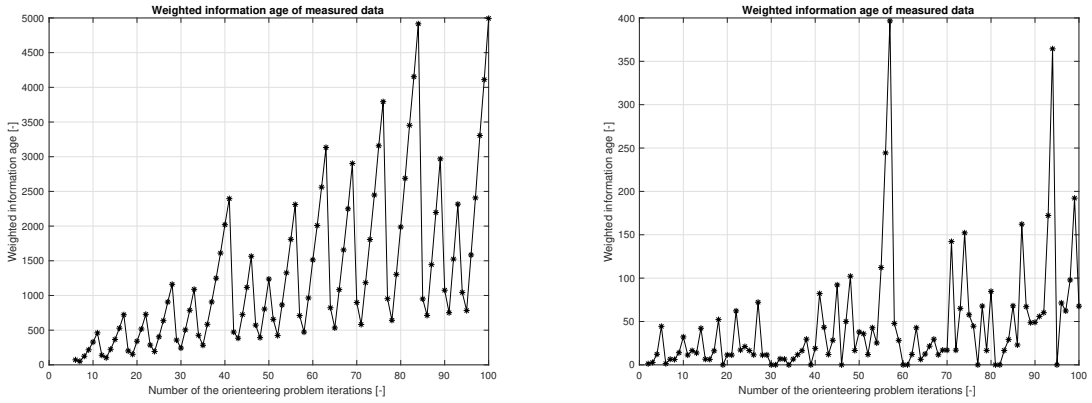
4.3 Orienteering problem

The sum of all measurement ages weighted by their variability is given as follows:

$$a_i = \sum_{n \in B(n)} u_n s_n,$$

where u_n is the number of iterations since the node n was lastly visited and s_n is its initial score value. The time step is represented as an iteration of the orienteering problem. The maximum time budget was chosen to be 450 s wich is a value lower than usual UAV flight times since some time should be reserved for actual measurements.

The time budget for the first simulation was chosen to be 300 s. Our simulation shows that with this time budget the UAV is not able to cover tho whole area and the weighted information age increases rapidly with time (see Figure 4.6a). When the time budget was chosen to be 450 s, the weighted information age of measured data still increases with time, but much slower (see Figure **??**). To get better results, the time budget should be even higher or more UAVs should be used.

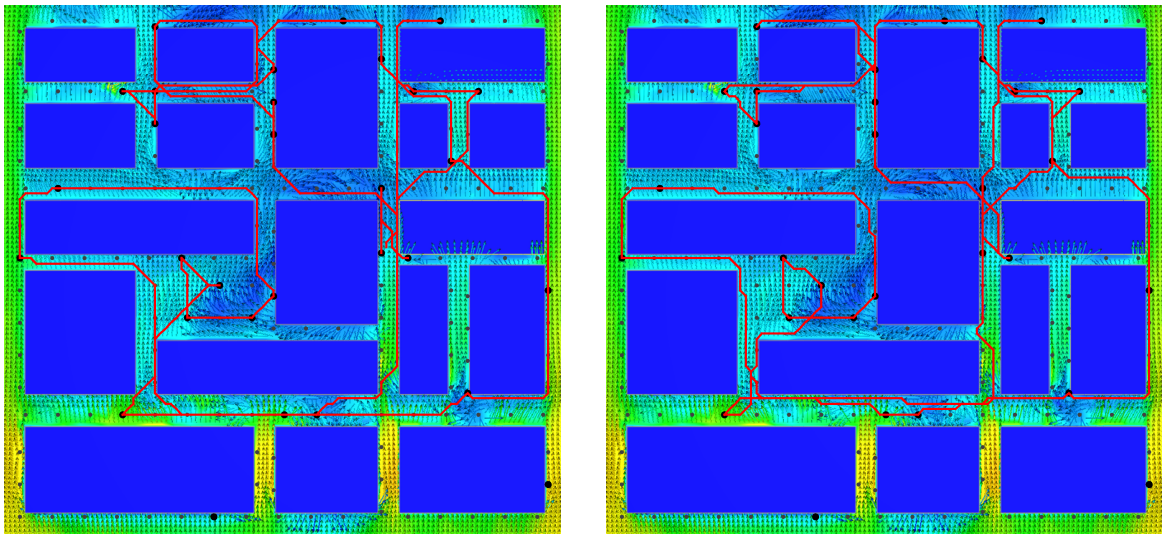


(a) time budget = 300 s (b) time budget = 450 s

Figure 4.6: Weighted information age of measured data in simulation

To compare the two A* planners, the trajectory through set of nodes given from the orienteering problem was planned. On the Figure 4.7a the trajectory of UAV was planned using the simple A* planner. UAV traveling this trajectory does not change its direction as the wind velocity changes as can be seen on the bottom part of the Figure. The travel time is this case was $t = 469$ s. On the Figure 4.7b the trajectory of UAV was planned using the wind A* planner. UAV traveling this trajectory uses the tailwinds and avoids the headwinds. The planned trajectory is longer, as can be seen for example on the bottom part of the Figure where the UAV travels close to buildings where the wind speed is lower. The

travel time in this case was $t = 449\ s$.



(a) Simple A* planner, $t = 469\ s$

(b) Wind A* planner, $t = 449\ s$

Figure 4.7: Planned trajectories using different A* planners.

Chapter 5

Conclusion

In this thesis the problem of the wind field measurement in the urban environment using the Unmanned Aerial Vehicles (UAVs) was discussed. The idea was inspired by The Small Business Innovation Research (SBIR) program, specifically the 'Microcosm Forecasting Utilizing Swarm Unmanned Aerial Vehicle Technology'. The aim was to design and implement method for periodical wind measurements in urban environnment using UAVs for improving CFD models which takes into account the wind field in given area and the maximum flight time of UAV. The algorithm ensures the UAV visits defined set of measuring points and minimizies the average weighted information age of measured data. The provided algorithm also ensures that the measuring points where the wind velocity has been changing more are visited sooner and more often than other points.

In the first part of this thesis the overview of the measuring sensors and state-of-the-art methods talking about same problem was provided. In the second part, methods for the optimal sample placement of measuring points and the path planner were designed and implemented. In the last part of this thesis, those methods were proved in simulation of one particular scenario of the urban environment model. The simulation used the CFD software to create the wind field.

For the future work, we would like to implement a method for more UVAs and compare the results from proposed iterative method with the results of the team orienteering problem solver. Also, the proposed method works only with one main direction of the wind, in this thesis the wind flowing from the south to the north. To the future, the generalization of this method for varying main wind direction will be needed. For this task the similar aproach to that of Du et al. [6] can be used.

Bibliography

[1] BAILEY, T. C. – GATRELL, A. C. *Interactive spatial data analysis*. 413. Longman Scientific & Technical Essex, 1995.

[2] BALLARI, D. – DE BRUIN, S. – BREGT, A. K. Value of information and mobility constraints for sampling with mobile sensors. *Computers & Geosciences*. 2012, 49, s. 102–111.

[3] BLOCKEN, B. – CARMELIET, J. Pedestrian wind environment around buildings: Literature review and practical examples. *Journal of Thermal Envelope and Building Science*. 2004, 28, 2, s. 107–159.

[4] CASTELLO, C. C. et al. Optimal sensor placement strategy for environmental monitoring using wireless sensor networks. In *System Theory (SSST), 2010 42nd Southeastern Symposium on*, s. 275–279. IEEE, 2010.

[5] CHAO, I.-M. – GOLDEN, B. L. – WASIL, E. A. A fast and effective heuristic for the orienteering problem. *European journal of operational research*. 1996, 88, 3, s. 475–489.

[6] DU, W. et al. Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs. In *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*, s. 167–178. IEEE, 2014.

[7] DUMAIS, R. E. et al. Performance Assessment of the Three-Dimensional Wind Field Weather Running EstimateNowcast and the Three-Dimensional Wind Field Air Force Weather Agency Weather Research and Forecasting Wind Forecasts, December 2012.

[8] ELSTON, J. et al. Overview of small fixed-wing unmanned aircraft for meteorological sampling. *Journal of Atmospheric and Oceanic Technology*. 2015, 32, 1, s. 97–115.

[9] FISCHETTI, M. – GONZALEZ, J. J. S. – TOTH, P. Solving the orienteering problem through branch-and-cut. *INFORMS Journal on Computing*. 1998, 10, 2, s. 133–148.

[10] GARAU, B. – ALVAREZ, A. – OLIVER, G. Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, s. 194–198. IEEE, 2005.

[11] GOLDEN, B. L. – LEVY, L. – VOHRA, R. The orienteering problem. *Naval research logistics*. 1987, 34, 3, s. 307–318.

[12] INC., A. *CFD*, 2016. Dostupné z: `https://www.autodesk.com/products/cfd/overview`.

[13] INC., A. Transient Parameters, December 2015. Dostupné z: `https://knowledge.autodesk.com/support/cfd/learn-explore/caas/CloudHelp/cloudhelp/2014/ENU/SimCFD/files/GUID-7BE54C33-D7CE-4F95-81B0-FD3B5517E9B3-htm.html`.

[14] INC., A. Wind Tunnel, February 2015. Dostupné z: `https://knowledge.autodesk.com/support/flow-design/learn-explore/caas/CloudHelp/cloudhelp/ENU/FlowDesign/files/GUID-9B85F4A0-5072-454D-8710-CCFF26507BE9-htm.html`.

[15] KONAK, A. Predicting coverage in wireless local area networks with obstacles using kriging and neural networks. *International Journal of Mobile Network Design and Innovation*. 2011, 3, 4, s. 224–230.

[16] LAPORTE, G. – MARTELLO, S. The selective travelling salesman problem. *Discrete applied mathematics*. 1990, 26, 2-3, s. 193–207.

[17] LEIFER, A. C. – ROSENWEIN, M. B. Strong linear programming relaxations for the orienteering problem. *European Journal of Operational Research*. 1994, 73, 3, s. 517–523.

[18] MILLER, C. E. – TUCKER, A. W. – ZEMLIN, R. A. Integer programming formulation of traveling salesman problems. *Journal of the ACM (JACM)*. 1960, 7, 4, s. 326–329.

[19] MOYANO CANO, J. Quadrotor UAV for wind profile characterization. Master's thesis, Universidad Carlos III de Madrid, Campus de Leganés, Avda. de la Universidad,30 28911 Leganés, Spain, 2013.

[20] PěCHOUčEK, M. – ROLLO, M. Informované metody prohledávání stavového prostoru. Department of Cybernetics, Czech Technical University in Prague.

[21] *Microcosm Forecasting Utilizing Swarm Unmanned Aerial Vehicle Technology.* The Small Business Innovation Research, 2015. http://www.acq.osd.mil/osbp/sbir/solicitations/sbir20161/index.shtml.

[22] SPIESS, T. et al. First application of the meteorological Mini-UAV'M2AV'. *Meteorologische Zeitschrift*. 2007, 16, 2, s. 159–169.

[23] TSILIGIRIDES, T. Heuristic methods applied to orienteering. *Journal of the Operational Research Society*. 1984, 35, 9, s. 797–809.

[24] VANSTEENWEGEN, P. – SOUFFRIAU, W. – VAN OUDHEUSDEN, D. The orienteering problem: A survey. *European Journal of Operational Research*. 2011, 209, 1, s. 1–10.

[25] WARE, J. – ROY, N. An analysis of wind field estimation and exploitation for quadrotor flight in the urban canopy layer. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, s. 1507–1514. IEEE, 2016.

[26] WREN, A. – HOLLIDAY, A. Computer scheduling of vehicles from one or more depots to a number of delivery points. *Journal of the Operational Research Society*. 1972, 23, 3, s. 333–344.

[27] YUE, Z. – MI, Z. Deductive method of wind field characteristics based on measured data. In *Electric Technology and Civil Engineering (ICETCE), 2011 International Conference on*, s. 210–213. IEEE, 2011.

Appendix A

Contents of the enclosed CD

Directory or file name	**Description**
BachelorThesis.pdf	this thesis in PDF format
bachelorThesis	latex source code of this thesis
source	Java source code
model	the model of urban environment created in the Autodesk AutoCAD for Mac 2017
CFD	created simulations in the Autodesk CFD 2017 software in selected time steps with exported datasets
python	python source code to create measuring points in the Autdoesk CFD 2017 software