Planning and Evaluation of UAV Mission Planner for Intralogistics Problems

Thiago R. F. Cavalcante
School of Electrical and Computer Engineering
Federal University of Amazonas
Amazonas, Manaus 69067-005
Email: thiagorodrigoengcomp@gmail.com

Iury V. de Bessa
Department of Electricity
Federal University of Amazonas
Amazonas, Manaus 69067-005
Email: iurybessa@ufam.edu.br

Lucas C. Cordeiro
Department of Computer Science
University of Oxford
Oxford, United Kingdom
Email: lucas.cordeiro@cs.ox.ac.uk

***Abstract*—This paper presents the development of mission planners in intralogistics for a commercial unmanned aerial vehicle equipped with a robotic gripper in an industrial environment, which consists of an input warehouse, production lines, and a product depot. In this study, the planner produces the needed commands for carrying out a given mission, which includes the delivery of inputs brought from the warehouse to the production line until the final product is delivered to the client (product depot). It was developed two different approaches for mission planning: in the first approach, a simple heuristic is used to solve the mission problem, where a UAV gets the necessary inputs to produce a product, from the warehouse, and bring to the respective production line and the UAV waits in production line site to finish the production of the product; in the second approach, a technique with task scheduling (production process) is employed; both approaches follow a set of production rules. An evaluation of the developed mission planners is performed, verifying the cost of both approaches, measuring the execution time, and comparing those results with the optimum cost obtained with the IBM ILOG CPLEX optimizer.**

I. Introduction

Logistics has become a competitive and fundamental factor for organizations, involving the management, conservation, and supervision of freight transport. In addition, excellent logistics means customer satisfaction; so speed is still an important factor in a successful logistics process [?drone4logistic]. Currently, one of the solutions to this type of problem is the use of unmanned aerial vehicles (UAVs). Nowadays, UAVs are mostly remotely piloted vehicles (RPV), since their operations are carried out by ground operators. If the tasks performed by a UAV are performed autonomously, it would relieve the work of these operators, since they perform tedious and repetitive tasks [?pascarella2013autonomic].

One possible improvement of these logistics systems is the increase of the UAVs automation, which results in costs minimization. Consequently, investments and studies related to stand-alone UAVs are important to the smart factories development [?hern2014dhl]. However, one of the main problems for using autonomous UAVs is the system's reliability and intelligence. Thus, increased employment of autonomous UAVs requires the development of devices, which are able to perform tasks and interact with the environment in an intelligent and reliable way.

Autonomous UAVs need to know what will happen in a future instant and what is the best decision to make at the present time; therefore, they require strategies not only to decompose their missions into meaningful sub-tasks, but also to track progress toward mission goals and the evolution of these tasks relative to the autonomous UAVs capabilities [?finn2012developments]. As a consequence, in order to successfully perform a mission, it is recommended to perform task planning [?finn2012developments]. Mission planning problems consist of planning events to meet certain requirements and objectives [?krozel1988search]. Therefore, planning events is one of the main challenges faced in solving this problem.

Both academy and industry have done researches about evaluation and optimization of mission planning in the last years. schwarz2012towards have used ant colony to optimize UAV missions. Another paper investigates energy consumption for a factory and evaluates the logistic planning processes using statistical metrics of evaluation [?muller2012analyzing].

To evaluate mission planning strategies, evaluation metrics must be employed. An evaluation metrics consists of a set of measures that follow a common underlying evaluation methodology. It is used to evaluate the efficacy of information retrieval systems and to justify theoretical and/or pragmatical developments of these systems [?pehcevski2009evaluation]. In this work, we use an optimal measure to compare with a calculated value of a mission cost.

This paper presents a methodology that evaluates the cost of mission planners for a commercial UAV. We developed an evaluation metrics that evaluates the relative cost of a planning strategy related with the optimal cost generated by the CPLEX optimizer [?cplex2003ilog]. In summary, the main contributions of this study are:

- a novel evaluation methodology for UAV intralogistics mission planners algorithms, which allows predicting

the planners performance and also obtaining optimal algorithms and missions;
- development of an intralogistcs mission planner framework that provides mission commands for a UAV system;
- use of a commercial UAV system in intralogistics missions to demonstrate the evaluation methodology efficiency.

As a result to this work, we have verified that the development and experiments of the mission planner evaluation methodology were done successfully as sown in Section VI. Additionally, the framework to command and control a UAV was gratefully developed and tested in simulated and real environment.

Outline. Section II shows previous studies related to mission planning, optimization, and evaluation. Section III provides the fundamentals of mission planning and optimization problems. Section IV describes the UAV movement system used in this work. Section V explains the proposed evaluation methodology in further details. Section VI describes the experimental procedures and results in order to explore and demonstrate the potential of methodology, and, finally, Section **??** concludes this study and describes future work.

II. Related Work

In the literature, there are attempts to implement UAV guidance systems that perform mission planning. doherty2009temporal presented a framework architecture for mission planning and execution tracking applied to an unmanned helicopter. During the mission execution, knowledge is acquired through sensors, which were used to create state structures. These structures allow constructing a logical model, representing real system development and environment over time. The planning and monitoring modules use temporal action logic (TAL) to reason about actions and changes.

The NASA/U.S. Army autonomous helicopter project has developed a guidance system for the autonomous surveillance planning problem for multiple and different targets [?whalley2005design], which generates mission plans using a theoretical approach for decision making. A high-level standalone control is provided by the framework Apex [?baer1998nasa], a reactive procedure-based scheduler/planner used to perform mission-level tasks. Apex synthesizes a course of action primarily by linking elemental procedures expressed in procedural definition language (PDL), a notation developed specifically for the Apex reactive planner. This guidance system is integrated into a robotic helicopter and tested in more than 240 scenarios.

A similar project, called Ressac (Research and Rescue by Cooperative Autonomous System), is conducted by the French Aerospace Laboratory (ONERA) for a search and rescue scenario [?fabiani2007autonomous]. This architecture for an exploration mission is developed based on the idea of decomposing the mission into a sequence of tasks or macro-actions associated with rewards. The problem is modeled using a Markov decision process framework (MDP) and dynamic programming algorithms for mission planning. Konigsbuch [?teichteil2007multi] extends the guidance system and integrates with a robotic helicopter.

The German Aerospace Center (DLR) has also developed a mission management system based on the behavior paradigm [?adolf2010onboard], which has been integrated with the ARTIS helicopter and validated in different scenarios, including waypoints follower, search and tracking mission.

rodriguezstudy investigate the performance analysis of UAV operators, w.r.t. agility, consumption, aggressiveness, precision and reflexes; each of those aspects has an evaluation metrics, in order to discover behavioral pattern of the operators.

Existing approaches for evaluation of mission planners for intralogistics problems are either empirical or theoretical. This paper describes an approach combining both aspects. In addition, we test the metrics in a real environment, *i.e.*, using a real UAV to perform a mission and we measure the cost (mission execution time) of the mission.

III. Preliminaries

A. Terminology

Key definitions related to the case study and the application developed in this study need to be clarified. All definitions below are adopted in the remainder of this study.

Definition 1: **(Mission Command)** Mission Command is a command created to execute a task such as to go from one location to another, get a package using a robot gripper, and land a UAV.

Definition 2: **(Mission)** Mission is the set of steps and mission commands that the UAV executes to produce the customer's order.

Definition 3: **(Warehouse)** Warehouse is the set of stored raw material available until the moment of entering the productive process. The raw materials, *i.e.*, the inputs available in this work are inputs A, B, and C.

Definition 4: **(Order)** Order is the requisition of products made by the client. In this study, the products are of type X and Y.

Definition 5: **(Production Time)** Production time is the time required to produce a product X or Y, after making available all the needed inputs for the production, given by the production rule.

Definition 6: **(Production Rule)** Production rule describes what and how many inputs are needed to produce a particular product.

Definition 7: **(Mission Planner)** Mission planner is the agent who performs the planning of a mission, that is, produce all steps and commands needed to carry out a given mission.

Definition 8: **(Mission File)** The mission file is a file that is created for the context of this work, with the extension .MISSION containing the mission itself.
Definition 9: **(Movement Function)** The movement function are functions created in Python using the drokekit API to send commands to the UAV by MAVLink protocol.

Definition 10: **(Production Mission)** The production mission is the set of steps to produce all the product required by the client.

B. Mission Planning

Firstly, a mission can be defined as a goal that needs to be completed (cf. Definition 2). In the context of this study, the UAV mission is the packages delivery according to a set of well defined rules. A definition to mission planning for UAV is the process of planning the locations to visit (waypoints) and the actions that the vehicles can perform (e.g., loading/dropping a load and taking videos/pictures), typically over a time period [`?ramirez2014solving`]. Functionally, mission planning lies above the trajectory planning process, where the mission planner (cf. Definition 7) generates a desired mission plan, and then the trajectory planner generates the flight plan (trajectories) between the waypoints.

C. Optimization Problems

An optimization problem is related to finding the best solution (relative to a certain criterion) among a set of available alternatives. For instance, the popular bin packaging problem that aims to find the number of boxes of a certain size to store a set of objects of indicated sizes; optimization involves, for example, finding the least amount of boxes. An optimization problem is usually represented as follows:

$$\begin{array}{ll} \min & f(\mathbf{x}), \\ \text{s.t.} & \mathbf{x} \in \Omega. \end{array} \tag{1}$$

where Ω is a set of the problem constraints and $f(x)$ is the function to optimize.
An optimization problem can be defined as a finite set of variables, where the correct values for the variables specify the optimal solution. If the variables are of the set of real, the problem is called continuous, and if they can only have a finite set of distinct values, the problem is called combinatorial [`?korte2012combinatorial`].

IV. UAV Movement System

The core hardware of the UAV IRIS+ is the Pixhawk and we can control it using a Python library [`?dronekit`], which uses Micro Air Vehicle Link (MAVLink) protocol [`?meier2011pixhawk`]. MAVLink is a protocol for communicating with small unmanned vehicle, which is designed as a header-only message marshalling library.
The IRIS+ UAV is integrated into a robot gripper to take and leave packages during missions (cf. Definition 2). We have connected a servo motor to the Pixhawk by one of the pulse width modulation (PWM) outputs. Figure 1 shows the system hardware architecture and the interconnections between each component module. In the hardware architecture shown in the Figure 1, we can see the UAV hardware component connections where there is the Pixhawk (flight controller) and its connections between other components such as the compass, GPS, PWM outputs, battery and etc. Moreover, it shows the connection with a robot gripper using a PMW outupts as a signal control for the servo motor in the robot gripper. Finally, it shows the communication between a personal computer (PC) and the UAV via radio control (RC) signal.

Command	Description
`TakeOff`	takes off the UAV
`GoTo`	moves the UAV to a certain location
`TakePackage`	takes an input/product (gripper)
`LeavePackage`	leaves an input/product (gripper)
`Wait`	makes the UAV to hover (wait)
`Land`	lands the UAV

TABLE I: Description of movement functions

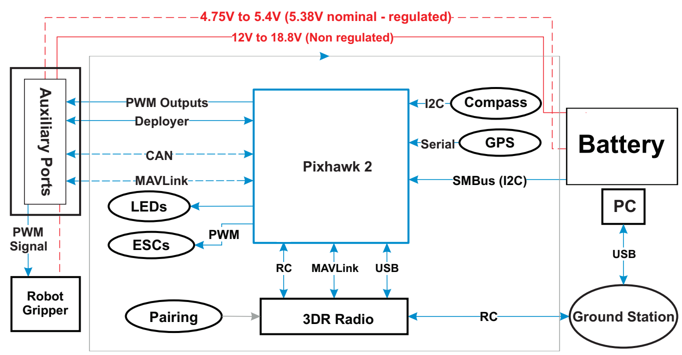

Fig. 1: System Hardware Architecture.

In the software architecture, the Mission Planner (cf. Definition 7) reads the warehouse inputs and client order and produces a `.mission` file, which contains the list of mission commands needed for producing the required client order. This `.mission` file is used by a UAV Control Program to control the UAV and to produce the low-level movement commands using MAVLink protocol (cf. Definition 1). Figure 2 shows the mission planning framework software components.

In order to control the UAV from a PC, we have used the dronekit API that translates MAVLink commands to a Python function. In the ground station, the PC is running the UAV Control Program that controls the UAV using a radio module connected to the PC via USB. We have created a bunch of functions in the control program for the most common UAV actions. The movement functions (cf. Definition 9) are described in Table I.

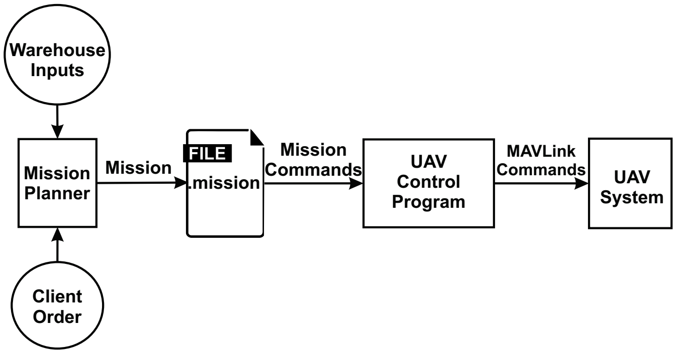

Fig. 2: System's Architecture.

V. Methodology of Time Cost Evaluation, UAV Use and Mission Planning

A. Case Study: UAV Intralogistics Mission

In order to model the mission planning problem as an optimization problem, the case study shown in Figure 3 is used.

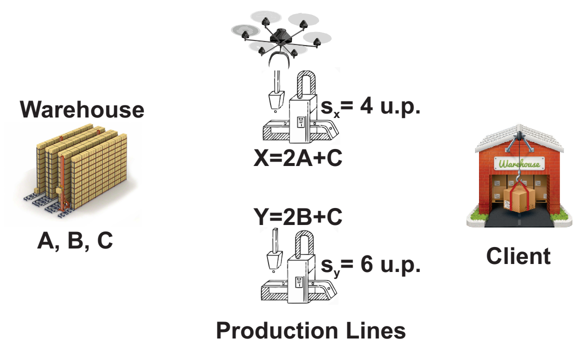

Fig. 3: Case Study Representation.

Figure 3 shows that there are three types of inputs in the warehouse (*i.e.*, A, B, and C) and two production lines that produce two different products (*i.e.*, X and Y). Each production line produces only one type of product and has a characteristic production time (cf. Definition 5). Figure 3 shows that to produce a product of type X, two inputs of type A and one input of type C are required, and to produce a product of type Y, two inputs of type B and one input of type C are required. The production time of a X product is $4p.u.$ and the time of production of product Y is $6p.u.$. A production unit ($1p.u.$) is considered to be a `GoTo` command performed by the UAV.
The task to be performed is the production of the client order (cf. Definition 4), where a given UAV collects supplies from the warehouse, takes that to the production line, and once the production of a certain product is finished, the UAV delivers it to the client.

B. Modeling a UAV Intralogistics Mission as an Optimization Problem

The purpose of this subsection is to make the mission planning problem into an optimization problem, creating a modeling for the problem; in order to find, afterwards, the shortest execution time of all tasks (minimization), based on the case study explained in Section V-A. The notation used is given below:

- $\mathcal{T} = \{T_j | j \in \mathbb{N}^*, j \leq N\}$ is the set of N tasks;
- $\mathcal{M} = \{m_i | i \in \mathbb{N}^*, i \leq M\}$ is the set of M production lines (machines);
- $\mathcal{P} = \{p_j | j \in \mathbb{N}^*, j \leq N\}$ is the processing time of each j-th task;
- $S = \{s_i | s \in \mathbb{N}^*, i \leq M\}$ is the setup (production) time of each i-th production line;

a) Decision variable: The variable x_{ij} is a binary decision variable that takes the value 1 if the task j is running on the machine i; and 0 otherwise. The variable $C_{mission}$ is the variable that we want to optimize.

$$x_{ij} = \begin{cases} 1, & \text{if the task } j \text{ is running in} \\ & \text{the machine (production line) } i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

b) Objective function: The objective function is the total mission cost $C_{mission}$ (total process execution time) that can be modeled as follows.

$$C_{mission} = \sum_{i=1}^{M} \sum_{j=1}^{N} (p_j + s_j) x_{ij}, \tag{3}$$

Eq. (3) represents the sum of the duration time p_j of each travel from one place to another in the case study explained in Section V-A, considering the production time (cf. Definition 5) s_j in each production line.

c) Constraints:

- Each task must be executed/processed in a unique machine:

$$\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} = 1 \tag{4}$$

- Execution time of each machine:

$$C_{mission} \leq C_{max} \tag{5}$$

Eq. (5) indicates that the mission cost is always less or equal than a maximum cost denoted by C_{max}, obtained empirically.

d) Resulting optimization problem: The resulting optimization problem consists in minimizing C_{max} w.r.t. the decision variable (2) constrained to the condition in Equations (4) and (5). Thus, the optimization problem is represented as follows:

$$\begin{aligned} \min \quad & C_{mission}, \\ \text{s.t.} \quad & \textstyle\sum_{i=1}^{M} \sum_{j=1}^{N} x_{ij} = 1, \\ & C_{mission} \leq C_{max} \end{aligned} \tag{6}$$

C. Planner Evaluation Methodology

The main contribution of this study is a methodology to evaluate UAV mission planner algorithms and to find minimum cost planner. For this purpose, a generalized evaluation metrics is developed. The objective (cost) function modeled in Section V-B is related to the total time spent for the mission execution. Our evaluation metrics compares the cost of a planner algorithm with the best cost computed by the CPLEX solver [?cplex2003ilog]. This work proposed a novel metrics called Mission Planner Cost Index ($MPCI$).

Firstly, the optimal cost of the problem is obtained by means of the CPLEX solver, which returns the optimum value (minimum mission execution time). The model proposed in Section V-B is implemented using the CPLEX solver library available for C++.

The cost of each planner strategy (Algorithms 1 and 2) c_X is obtained by counting the number of `GoTo` commands which represents a process (task).

Finally, the evaluation of each mission planner is computed w.r.t. the optimal cost, therefore, the $MPCI_X$ of a planner X is computed as follows:

$$MPCI_X = \frac{c_o}{c_X}, \quad (7)$$

where c_o is the optimal cost obtained by the CPLEX solver, c_X is the cost of the solution generated by planner X, and $0 \leq MPCI_X \leq 1$. Note that as close to 1 the $MPCI_X$ is, the solution cost becomes smaller.

D. Mission Planners

In this study, we considered that mission planner is a software that generates a production mission (cf. Definition 10) given the warehouse and customer order. This program generates a `.mission` extension file containing a set of mission commands (cf. Definition 1), as described in Section IV. Two examples of planners are presented in this work and are employed to demonstrate the cost evaluation methodology.

1) Planner A: In Algorithm 1, we show a strategy to solve the mission planning problem and we denoted it as Planner A. In this particular algorithm, the production of X products has a higher priority over Y, *i.e.*, the inputs are firstly allocated to production of X orders, and the production of Y products begins if there is no other X to be produced. The general steps of planner A are described in Algorithm 1.

Algorithm 1 Planner A

```
Input: warehouse
Input: order
Output: mission file .mission
begin
    check the order;
    repeat
        go to the warehouse;
        repeat
            get input A;
            bring to the production line X;
        until until bring 2 A elements;
        go to the warehouse;
        get the input C;
        bring to the production line X;
        wait X to be produced;
        bring X to the client;
    until production of all X elements finish;
    repeat
        go to the warehouse;
        repeat
            get input B;
            bring to the production line Y;
        until until bring 2 B elements;
        go to the warehouse;
        get the input C;
        bring to the production line Y;
        wait Y to be produced;
        bring Y to the client;
    until production of all Y elements finish;
end
```

2) Planner B: The strategy for Planner B is a bit more complex than Planner A. In the Planner B strategy, the UAV starts to bring all the necessary inputs to make the first product X, taking all the A, B and C inputs, respectively, to the production line X. After bring all the necessary inputs to produce the first X product, the production line X starts to produce the X product and while the production line X is producing, the UAV goes to the warehouse to get the necessary inputs to produce the next product (either Planner A and Planner B produce firstly the X products e then all the Y products). However, when the X product finishes producing, the UAV knows the instant and goes to the production line to get the X product to bring to the client place; and after that, the UAV goes back to bring the rest of the inputs. The UAV keeps work in the same way until it brings all the products to the client place. Differently to the Planner A, the Planner B does not wait the production in the production line. The UAV works as a scheduler and executes the mission faster than Planner A strategy, because it does not enter in a busy wait state. The general steps of planner B are shown in the Algorithm 2.

VI. Experimental Evaluation

This section describes the experimental results obtained in this project, as well as the cost evaluation of two techniques used, which are compared to the optimum cost implemented with the CPLEX solver.

A. Experimental Environment and Objectives

In order to verify the efficiency of the metrics shown in Section V, our experimental evaluation aims to answer the following research questions:

RQ1 Does the framework for mission planning, command and control for intralogistics mission using a UAV produce the expected results?

RQ2 Is the metrics of mission evaluation efficient?

The mission planning algorithms were executed in a computer running Linux Mint OS, core i7 processor and 8 GB of RAM. In order to control the UAV, we run the control program, which uses the dronekit API as an interface

Algorithm 2 Planner B

```
Input: warehouse
Input: order
Output: mission file .mission
begin
    initialize t_x;
    initialize t_y;
    check the order;
    repeat
        if the counter of that X is not t_x then
            go to the warehouse;
            repeat
                get the input A;
                bring to the production line X;
            until until bring 2 A elements;
            go to the warehouse;
            get the input C;
            bring to the production line X;
            start the counter of this X (production time);
            keep producing;
        else
            go back to the production line X;
            bring X to the client;
            go back to producing;
        end
    until production of all X elements finish;
    repeat
        if the counter of that X is not t_y then
            go to the warehouse;
            repeat
                get the input B;
                bring to the production line Y;
            until until bring 2 B elements;
            go to the warehouse;
            get the input C;
            bring to the production line Y;
            start the counter of this Y (production time);
            keep producing;
        else
            go back to the production line Y;
            bring X to the client;
            go back to producing;
        end
    until production of all X elements finish;
end
```

between a high level program language (Python) and the protocol that the UAV understands (MAVLink), in the same computer where there is a radio module connected via USB communicating with the UAV radio module.

B. Cost Evaluation

The results of each mission planner is compared to the optimal solution obtained with the branch-and-cut algorithm of the IBM/ILOG CPLEX 12.4 tool developed in C++ [?cplex2003ilog]. In order to obtain better results to perform the comparison, it was considered only the time in which the UAV takes to finish the production of a product.

	Planner A	Planner B	CPLEX
Time (s)	420	404	134

TABLE II: Planners and optimal (CPLEX) times

The Table II shows the mission execution times obtained using the planner algorithms A and B, and the minimum value provided by CPLEX. Using the metrics shown in the section V, then:

$$MPCI_A = \frac{134}{420} = 0.319 \quad (8)$$

$$MPCI_B = \frac{134}{408} = 0.328 \quad (9)$$

The MPCI indicates (cf. Eqs. (1) and (2)) that planner B performs the mission more quickly and has a lower cost than planner A.

C. Practical Results

In order to verify the practical results, as well as a cost comparison between the different approaches of mission planners developed in this work, the flight time measurement was performed using the two mission planning algorithms developed, using the case study shown in V-A. The experiments were performed in both, simulator and real UAV system. The simulations are performed with DroneKit-SITL [?dronekit], that is an resource of Dronekit Pythn API that allows to simulate the behavior and movement of a plane, a copter, or a rover, without the hardware, *i.e.* a real UAV. Additional experiments are performed with the real UAV system (3DR IRIS+). Figure 4 shows the map of experimental environment (Faculty of Physical Education and Physiotherapy of Federal University of Amazonas).

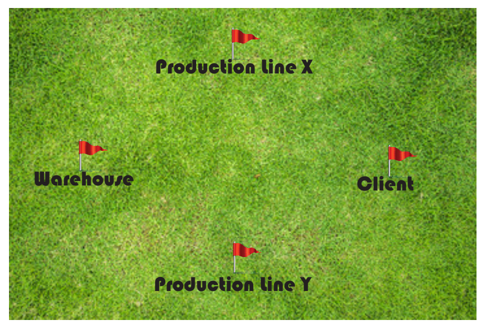

Fig. 4: Warehouse, Production Line X, Production Line Y and Costumer in the Map.

Table III presents the performance (total flight time) of both planners algorithms for simulations and tests with the (real) 3DR Iris+ UAV.

Test #	Flight Time of Planners			
	Simulator		3DR Iris+	
	A	*B*	*A*	*B*
1	460.41	436.08	455.12	441.72
2	460.69	436.89	456.93	440.18
3	460.08	441.68	457.19	447.51
4	460.72	441.03	460.25	438.19
5	460.23	451.87	459.47	445.85

TABLE III: Mission Planners Flight Times.

Table III indicates that the mission time of planner B is lower then the time of planner A in all five tests with simulator and 3DR Iris+, ensuring the results of the evaluation methodology.

The aforementioned results confirmed the prediction provided by the planner evaluation methodology, and the planner B is faster than planner A in all the tests with simulations and IRIS+ tests.

D. Threats to Validity

In order to compile the experiments, we have assembled a favourable environment to apply our evaluation metrics. In this way, we considered the use case in Section V.
However, in case we change the scenario where the number of UAVs increases, our algorithms will not work as expected because we did not adapted the algorithms for cooperative work and consequently our metrics will not work as expected. Further works may be implemented to make the metrics works in a cooperative work environment where the number of UAVs is greater than two.

VII. Conclusion

The conclusion goes here.

Acknowledgment

The authors would like to thank...