Planning and Evaluation of UAV Mission Planner for Intralogistics Problems

Thiago R. F. Cavalcante*, Iury Bessa*, Lucas C. Cordeiro*†
*Federal University of Amazonas, Manaus, Brazil
†University of Oxford, Oxford, United Kingdom
Email: thiagorodrigoengcomp@gmail.com, {iurybessa,lucascordeiro}@ufam.edu.br

***Abstract*—We describe and evaluate 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 particular study, the planner produces the needed commands for carrying out a given mission, which includes the delivery of inputs picked up from the warehouse to the production line until the final product is delivered to the client (product depot). We propose two different approaches for mission planning: in the first approach, a simple heuristic is used to solve the mission problem, where a UAV obtains the needed inputs to produce a product from the warehouse, and then it brings the product to the respective production line and waits to finish its production; in the second approach, a technique with task scheduling (production process) is employed; both approaches follow a set of production rules. In addition, a novel evaluation methodology for mission planner algorithms is proposed in order to verify the cost of both approaches, measure the execution time, and compare 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 client satisfaction; so speed is still an important factor in a successful logistics process [1]. Currently, one of the solutions to this type of problem is the use of unmanned aerial vehicles (UAVs) or commonly known as drones. Nowadays, UAVs are mostly remotely piloted vehicles (RPV), since their operations are carried out by ground operators. If the UAV could perform the tasks autonomously, then it would relieve the work of those operators, since they perform tedious and repetitive tasks [2].

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 [3]. However, one of the main problems for using autonomous UAVs is the system 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 [2].

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 subtasks but also to track progress towards mission goals as well as the evolution of those tasks with respect to the autonomous UAVs capabilities [4]. As a consequence, in order to successfully perform a mission, task planning is usually employed [4]. Mission planning problems consist of planning events to meet certain requirements and objectives [5]. Therefore, the event planning is one of the main challenges faced in solving this problem due to the mission complexity in some particular UAV applications.

Both academy and industry have conducted studies related to the evaluation and optimization of mission planning in the last years. Schwarz and Sauer [6] employ the ant colony algorithm to optimize UAV missions. Muller et al. [7] investigate energy consumption for a factory and evaluate the logistic planning processes using statistical metrics of evaluation.

To evaluate mission planning strategies, evaluation metrics must be employed. Evaluation metrics consist 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 development of those systems [8]. In this particular study, we use an optimal measure to compare with a pre-computed value of a mission cost.

Specifically, this paper presents a methodology that evaluates the cost of mission planners for a commercial UAV. We have developed an evaluation metric that checks the relative cost of a planning strategy related to the optimal cost generated by the CPLEX optimizer [9]. In summary, the main original contributions of this paper are:

- a novel evaluation methodology for UAV intralogistics mission planners algorithms, which allows predicting the planner's performance and also obtaining optimal algorithms and missions;
- development of an intralogistics 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 in both simulated and real environment.

Outline. Section II describes 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 study. 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 methodology benefits, and, finally, Section VII concludes this study and describes future work.

II. Related Work

In the associated UAV literature, there are attempts to implement UAV guidance systems that perform mission planning. Doherty et al. [10] present a framework architecture for mission planning and execution tracking applied to an unmanned helicopter. During the mission execution, knowledge is acquired through sensors, which are used to create state structures. These structures allow constructing a logical model, representing a 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 [11], which generates mission plans using a theoretical approach for decision making. A high-level standalone control is provided by the framework Apex [12], 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 Research and Rescue by Cooperative Autonomous System (Ressac for short), is conducted by the French Aerospace Laboratory (ONERA) for a search and rescue scenario [13]. This architecture (targeted at 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 and Fabiani [14] extend this guidance system and integrate that with a robotic helicopter.

The German Aerospace Center (DLR) has also developed a mission management system based on the behavior paradigm [15], which has been integrated with the ARTIS helicopter and validated in different scenarios, including waypoints follower, search and tracking mission.

Rodríguez-Fernández et al. [16] investigate the performance analysis of UAV operators, with respect to agility, consumption, aggressiveness, precision and reflexes; each of those aspects has evaluation metrics, in order to discover the behavioral pattern of the operators.

Recently, some works indicated the potential of the applications of drones for logistics operation, since these vehicles produce efficient and flexible routes [3] and reduce carbon dioxide emissions compared to emissions from truck-based logistics [17].

Tavana et al. [18] propose a bi-objective multi-product combined cross-docking truck-scheduling model with direct drone shipping and multiple fleets. The problem involved in that work is modeled using multi-objective mixed integer mathematical programming (MIP). Two opposite objective functions, *i.e.*, the total cost of allocation and scheduling and the scheduling time, are optimized concurrently.

Bae et al. [19] introduce an approach to investigate open storage yard using UAV and radio-frequency identification (RFID). The approach reduces the mismatch between information stored in inventory system and real-world data, the labor cost of investigating stock and equipment cost for the system.

Existing approaches for evaluation of mission planners for intralogistics problems are either empirical or theoretical. In contrast to previous studies, this paper describes and evaluates a novel approach by combining both aspects. In addition, we test the evaluation metrics in a real environment, *i.e.*, using a real UAV system to perform a mission and we then measure the cost (mission execution time) of that specific mission.

III. Preliminaries

A. Terminology

In this section, we describe the key definitions related to the case study which will be used throughout the paper.

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 client order.

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

Definition 4: **(Order)** Order is the product requirements placed 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 pre-defined 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, produces 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 represents a set of functions created in the Python programming language using the Drokekit Application Programming Interface (API) to send commands to the UAV by MAVLink protocol.

Definition 10: **(Production Mission)** The production mission is the set of steps needed to produce the product placed 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 package's delivery, according to a set of well-defined rules. A definition of 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 [20]. Functionally, mission planning lies above the trajectory planning process, where the mission planner (cf. Definition 7) generates the desired mission plan, and then the trajectory planner generates the flight plan (trajectories) between the waypoints.

C. Optimization Problems

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 smallest amount of boxes. An optimization problem is usually represented as follows [21].

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

where Ω is a set of the problem constraints and $f(x)$ is the function to optimize [21].

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 from the set of real numbers ($\mathbb{R}$), the problem is called continuous, and if they can only have a finite set of distinct values, the problem is called combinatorial [22], [23], [24].

To solve the optimization problems, it is necessary to develop a method that solves them, which are the algorithms. An important category of problems are the NP-hard problems, where they can only be solved by certain algorithms that try to arrive at the optimal solution of that determined problem. When the optimal solution of a NP-hard problem is not guaranteed, this type of method is called a heuristic. A heuristic is an intuitive way of solving a particular problem, where the best possible solution is not guaranteed [25].

Every optimization problem is basically characterized by having an objective function, which can be called cost function when it is desired to minimize it or utility function when it is desired to maximize it, and a set of constraints that delimit the space of viable solutions, or it is the region where the solutions are that can be accepted. The objective function contains a set of variables to which values must be assigned in a systematic way so as to walk through the search space and find the one that optimizes the result to be searched, in case a maximization problem finds the highest possible value while in a minimize the value. In both cases, the solution must meet the set of constraints imposed to be accepted

1) Scheduling Theory: Scheduling problems can be defined as the designation or allocation of certain resources to certain activities as a function of time [26]. Such allocation over time involves a decision-making process that aims to optimize one or more performance measurement criteria. Due to the fact that one of the first problems modeled as scheduling involves the optimization of the production of a factory, it became a consensus to define in a general way a scheduling problem as the allocation of tasks to machines or processors in such a way to optimize some criterion of production. These criteria are modeled as a mathematical function of maximization or minimization called the objective function.

There are many scheduling problems, varying according to the processing time, types and amount of constraints of the elements to be scaled, execution conditions and optimization criteria, *e.g.*, the number of machines can vary and they can be the same or different, the elements may or may not have a well-defined start and finish time, there may or may not be interruption in tasks (preemption) [27].

IV. UAV Movement System

In this section, we first investigate the UAV platform used (3DR IRIS+) in this study, describing the hardware characteristics and the control framework developed for intralogistics missions.

The core hardware of the UAV IRIS+ is the Pixhawk which is an independent, open-hardware project aiming at providing high-end autopilot hardware to the academic, hobby and industrial communities at low costs and high availability; it provides hardware for the Linux Foundation DroneCode Project; and we can control it using a Python library [28], which uses Micro Air Vehicle Link (MAVLink) protocol [29]. MAVLink is a protocol for communicating with a small unmanned vehicle, which is designed as a header-only message marshaling library.

The IRIS+ UAV is integrated into a robot gripper to obtain and release packages during the missions (cf. Definition 2). We have connected a servo motor to the Pixhawk

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

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 illustrated in 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 output, and battery. Moreover, it shows the connection with a robot gripper using a PWM output 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.

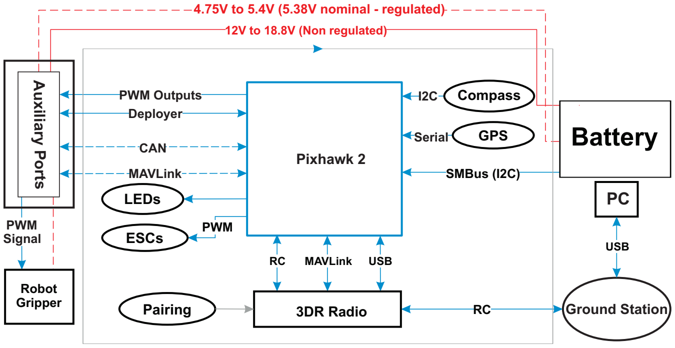

Fig. 1: System Hardware Architecture.

In the software architecture, the Mission Planner (cf. Definition 7) reads the warehouse inputs and client order and produces a `.mission` file (cf. Definition 8), 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 guide 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 several functions in the control program for the most common UAV actions. The movement functions (cf. Definition 9) are described in Table I.

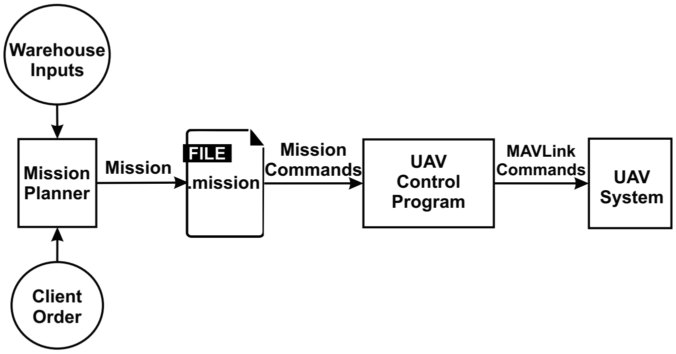

Fig. 2: System Architecture.

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

In this section, we have developed an optimization problem model of the problem of this work, evaluate two algorithm mission cost and describe a case study to a UAV and contents about mission planning.

A. *Case Study: UAV Intralogistics Mission*

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

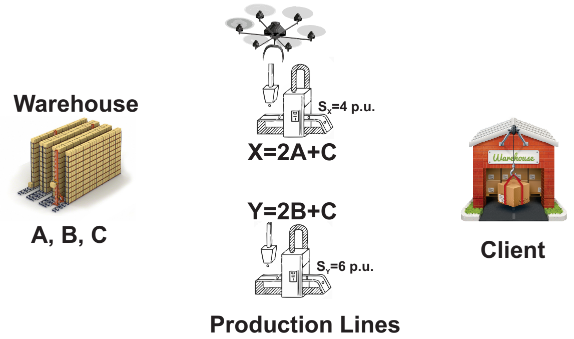

Fig. 3: Case Study Representation.

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, in order 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 production time 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 concluded, the UAV delivers it to the client.

B. Modeling a UAV Intralogistics Mission as an Optimization Problem

In this subsection, the aforementioned intralogistics mission planning problem is written as an optimization problem, by creating a mathematical model for the problem, in order to find, afterwards, the shortest execution time of all tasks (minimization), based on the case study described in Section V-A. The notation used here 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 set of processing times p_j of each j-th task;
- $S = \{s_i | s \in \mathbb{N}^*, i \leq M\}$ is the set of setup (production) times s_i 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.

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

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} with respect to the decision variable (2) constrained to the condition in (4) and (5). Thus, the optimization problem is represented as follows:

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

C. Planner Evaluation Methodology

The main contribution of this study is a methodology to evaluate UAV mission planner algorithms and to find a minimum planner cost. With this goal in mind, a generalized evaluation metrics must be developed. The objective (cost) function modeled in subsection 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 [9]. Here, we proposed a novel metrics scheme 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 subsection 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 with respect to the optimal cost, therefore, the $MPCI_X$ of a planner X is computed as follows:

$$MPCI_X = \frac{c_o}{c_X}, \tag{7}$$

where c_o is the optimal cost obtained by a C++ program that converts a given instance to a file in a format known by CPLEX, such as the LP format, *i.e.* the developed algorithm generates the model to be solved by CPLEX with the help of the mathematical programming tool UFFLP [30]. 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 client 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 1: In Algorithm 1, we show a strategy to solve the mission planning problem and we denoted it as Planner 1. In this particular algorithm, the production of X products has a higher priority over Y, i.e., the inputs are firstly allocated to the 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 1 are described in Algorithm 1.

As we can see in Algorithm 1, the method first check the order of the client to start planning the mission. The second step is to start the production firstly by bringing all the necessary inputs of the first product to the product line of that product. After that, the UAV will wait the product be produced (in p.u.) and then take the ready product to the client. The planner does that for all the product until it finishes the production process.

Algorithm 1 Planner 1

```
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 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 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 2: The strategy for Planner 2 is a bit more complex than Planner 1. In the Planner 2 strategy, the UAV starts to bring all the necessary inputs to make the first X product, 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 1 and Planner 2 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 1, the Planner 2 does not wait for the production in the production line. The UAV works as a scheduler and executes the mission faster than Planner 1 strategy because it does not enter into a busy wait state. The general steps of planner 2 are shown in the Algorithm 2.
Where t_X and t_Y are the production time (cf. Definition 5) of the production lines X and Y, respectively.

Algorithm 2 Planner 2

```
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 Y 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 Y to the client;
            go back to producing;
        end
    until production of all Y elements finish;
end
```

E. Correctness of the Optimization Model

In the direction of explaining the correctness of the model developed in subsection V-B, we describe the reasons below:

- we used a binary linear programming problem to build the model because of its characteristics (Eq. 4);
- only one product will be produced at time in a production line, this is a requirement in our problem and it is modeled in Eq. 4;
- the coefficients for the constraints are considered in Eq. 3;
- we considered that the execution time of a mission must be less or equal to a measured time obtained experientially (Eq. 5).

VI. Experimental Evaluation

This section describes the experimental results obtained in this project, as well as the cost evaluation of two techniques used, which is compared to the optimum cost implemented with the CPLEX solver[1].

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 are executed on 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 between a high level program language (Python) and the protocol that the UAV understands (MAVLink), on 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 are compared to the optimal solution obtained with the branch-and-cut algorithm of the IBM/ILOG CPLEX 12.4 tool developed in C++ [9]. In order to obtain better results to perform the comparison, it is considered only the time in which the UAV takes to finish the production of a product.

	Planner 1	**Planner 2**	**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 1 and 2, and the minimum value provided by CPLEX. Using the metrics shown in the section V, then:

$$MPCI_1 = \frac{134}{420} = 0.319 \tag{8}$$

$$MPCI_2 = \frac{134}{408} = 0.328 \tag{9}$$

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

C. Practical Results

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 is measured for two mission planning algorithms developed, using the case study shown in V-A.

The experiments are performed in both, simulator and real UAV system. The flight time is measured by using DroneKit-SITL [28], that is a feature of Dronekit Python API that allows simulating 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 the experimental environment (Faculty of Physical Education and Physiotherapy of Federal University of Amazonas).

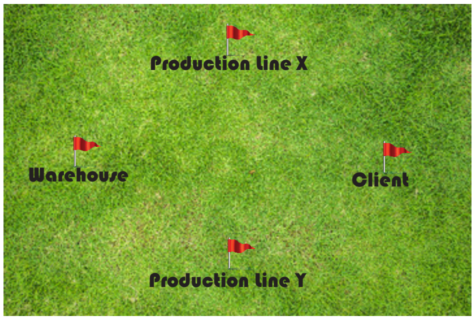

Fig. 4: Warehouse, Production Line X, Production Line Y and Costumer in the Map.

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+**	
	1	***2***	***1***	***2***
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 2 is lower than the time of planner 1 in all five tests with the 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 2 is faster than planner 1 in all the tests with simulations and IRIS+ tests.

D. Threats to Validity

In order to perform our experiments, a suitable environment is assembled to apply our evaluation metrics. In this way, we consider the case study described in Section V. Nevertheless, this case study is limited within the scope of this paper and the performance of our proposed approaches needs to be assessed on a larger benchmark set in future work. Additionally, in case we change the scenario, where the number of UAVs increases, our algorithms will not work as expected since we did not adapt those algorithms for cooperative work and consequently our metrics will not produce the expected results. In future work, we should also expand the evaluation metrics to work in a cooperative scheme environment, where the number of UAVs is greater than two.

[1] The code implemented in C++ using CPLEX library can be found on GitHub at https://github.com/thiago-cavalcante/optUAVCplex.git

VII. Conclusion

We have described an evaluation methodology for UAV mission planner in an industrial production scenario. In particular, we have used that evaluation methodology to evaluate the performance of two different algorithms. Our proposed approach uses an optimization tool to generate an optimal cost for a mission (with respect to the flight time) and then compares those algorithms using the Mission Planner Cost Index.

In addition, we have developed a framework for mission planning and control for intralogistics mission using a commercial UAV. We used this UAV to solve intralogistics problems using the Dronekit API for control purposes via a high-level programming language.

Our experiments were performed in both simulated and real environment using a commercial UAV to evaluate the mission planner algorithms for intralogistics problems. Moreover, our framework can solve the need for controlling a commercial UAV and thus contributes to a new approach to verify specific intralogistics problems.

Future work includes the use of computational vision for the recognition of inputs and improvements of the optimization problem modeling for better results in cost evaluation. Additionally, we will perform experiments in a cooperative work environment, where the number of UAVs is greater than two. In order to improve our results, we will develop more planner strategies such as an algorithm that produces different types of products simultaneously.

Aknowledgments

This paper is based on research sponsored by FAPEAM, CNPq and CAPES.

References

[1] Service-drone, "Drones for logistic and transport | multirotor," 2014, accessed in: 12/02/2016. [Online]. Available: http://www.service-drone.com/en/production/logistics-and-transport

[2] D. Pascarella, S. Venticinque, and R. Aversa, "Autonomic agents for real time UAV mission planning," in *IEEE 10th International Conference on Autonomic and Trusted Computing*. IEEE, 2013, pp. 410–415.

[3] A. Hern, "DHL launches first commercial drone 'parcelcopter'delivery service," *The Guardian*, 2014.

[4] A. Finn and S. Scheding, *Developments and challenges for autonomous unmanned vehicles*. Springer, 2012.

[5] J. A. Krozel, "Search problems in mission planning and navigation of autonomous aircraft," Tech. Rep., 1988.

[6] C. Schwarz and J. Sauer, "Towards decentralised agv control with negotiations." in *STAIRS*, 2012, pp. 270–281.

[7] E. Müller, H. Hopf, and M. Krones, "Analyzing energy consumption for factory and logistics planning processes," in *IFIP International Conference on Advances in Production Management Systems*. Springer, 2012, pp. 49–56.

[8] J. Pehcevski and B. Piwowarski, "Evaluation metrics for structured text retrieval," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 1015–1024.

[9] S. CPLEX, "Ilog," *Inc., Armonk, NY*, 2003.

[10] P. Doherty, J. Kvarnström, and F. Heintz, "A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 3, pp. 332–377, 2009.

[11] M. Whalley, M. Freed, R. Harris, M. Takahashi, G. Schulein, and J. Howlett, "Design, integration, and flight test results for an autonomous surveillance helicopter," in *Proceedings of the AHS International Specialists' Meeting on Unmanned Rotorcraft*, 2005.

[12] J. Baer-Riedhart, "Nasa environmental research aircraft and sensor technology program," *Aerospace America, January*, 1998.

[13] P. Fabiani, V. Fuertes, A. Piquereau, R. Mampey, and F. Teichteil-Königsbuch, "Autonomous flight and navigation of vtol uavs: from autonomy demonstrations to out-of-sight flights," *Aerospace Science and Technology*, vol. 11, no. 2, pp. 183–193, 2007.

[14] F. Teichteil-Königsbuch and P. Fabiani, "A multi-thread decisional architecture for real-time planning under uncertainty," in *3rd ICAPS'07 Workshop on Planning and Plan Execution for Real-World Systems*, 2007.

[15] F. Adolf and F. Andert, *Onboard mission management for a VTOL UAV using sequence and supervisory control*. INTECH Open Access Publisher, 2010.

[16] V. Rodríguez-Fernández, H. D. Menéndez, and D. Camacho, "A study on performance metrics and clustering methods for analyzing behavior in uav operations," *Journal of Intelligent & Fuzzy Systems*, no. Preprint, pp. 1–13, 2016.

[17] A. Goodchild and J. Toy, "Delivery by drone: An evaluation of unmanned aerial vehicle technology in reducing CO_2 emissions in the delivery service industry," *Transportation Research Part D: Transport and Environment*, 2017.

[18] M. Tavana, K. Khalili-Damghani, F. J. Santos-Arteaga, and M.-H. Zandi, "Drone shipping versus truck delivery in a cross-docking system with multiple fleets and products," *Expert Systems with Applications*, vol. 72, pp. 93–107, 2017.

[19] S. M. Bae, K. H. Han, C. N. Cha, and H. Y. Lee, "Development of inventory checking system based on UAV and RFID in open storage yard," in *International Conference on Information Science and Security (ICISS)*. IEEE, 2016, pp. 1–2.

[20] C. Ramirez-Atencia, G. Bello-Orgaz, M. D. R-Moreno, and D. Camacho, "Solving uav mission planning based on temporal constaint satisfaction problem using genetic algorithms," in *Proceedings of the 20th International Conference on Principles and Practice of Constraint Programming*, 2014, pp. 65–71.

[21] O. L. Mangasarian, *Nonlinear programming*. SIAM, 1994.

[22] A. Torn and A. Zilinskas, *Global Optimization*. New York, NY, USA: Springer-Verlag New York, Inc., 1989.

[23] J. Hu, Y. Wang, E. Zhou, M. C. Fu, and S. I. Marcus, *A Survey of Some Model-Based Methods for Global Optimization*. Boston: Birkhäuser Boston, 2012, pp. 157–179.

[24] D. R. Jones, "A taxonomy of global optimization methods based on response surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, Dec 2001.

[25] D. S. Hochbaum, *Approximation algorithms for NP-hard problems*. PWS Publishing Co., 1996.

[26] A. Burns and A. J. Wellings, *Real-time systems and their programming languages (2. ed.)*. Addison-Wesley, 1997.

[27] M. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, ser. SpringerLink : Bücher. Springer New York, 2012.

[28] I. 3D Robotics. Dronekit. [Online]. Available: dronekit.io

[29] L. Meier, J. Camacho, B. Godbolt, J. Goppert, L. Heng, M. Lizarraga *et al.*, "Mavlink: Micro air vehicle communication protocol," *Online]. Tillgänglig: http://qgroundcontrol.org/mavlink/start.[Hämtad 2014-05-22]*, 2013.

[30] A. A. Pessoa and E. U. Barboza, "Ufflp - an easy api for mixed, integer and linear programming," 2011, accessed in: 21/5/2017. [Online]. Available: http://www.gapso.com.br/ufflp/