# PLANNING AND UAV MISSION EVALUATION FOR INTRALOGISTICS PROBLEM

Thiago Cavalcante*, Iury Bessa†

*Graduate Program in Electrical and Computer Engineering, Federal University of Amazonas, Manaus, AM, Brazil

†Department of Electricity, Federal University of Amazonas, Manaus, AM, Brazil

Emails: `thiagorodrigoengcomp@gmail.com, iurybessa@ufam.edu.br`

**Resumo**— Este artigo apresenta o desenvolvimento de planejadores de missão na intralogística para um veículo aéreo não tripulado comercial, equipado com uma garra robótica, em um ambiente industrial onde há almoxarifado de insumos, linhas de produção e deposito de produtos. Neste trabalho, o planejador gera comandos necessários para realizar uma missão a qual compreende desde a entrega de insumos trazidos do almoxarifado à linha de produção, até a entrega do produto final ao cliente. Foram desenvolvidas duas abordagens diferentes para planejamento de missão: na primeira abordagem, utilizou-se uma simples heurística que resolve o problema; já na segunda abordagem, utilizou-se uma técnica com escalonamento de tarefas (processo de produção). Estas abordagens seguem algumas regras de produção que serão apresentadas ao longo deste trabalho. Foi realizado uma avaliação dos planejadores de missão desenvolvidos, verificando o custo de ambos, realizando algumas medidas de tempo de execução, bem como comparando estes resultados com o custo ótimo obtido com a ferramenta de otimização CPLEX.

**Palavras-chave**— Planejamento de Missão, Sistemas de Manufatura.

**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 where there are a warehouse of inputs, production lines and a product warehouse. In this work, the planner generates the necessary commands to carry out a mission that includes everything from the delivery of inputs brought from the warehouse of inputs to the production line until the final product is delivered to the customer (product warehouse). Two different approaches were developed for mission planning: in the first approach, a simple heuristic was used to solve the problem; in the second approach, a technique with task scheduling (production process) was used. These approaches follow some production rules that will be presented throughout this work. An evaluation of the mission planners developed was performed, verifying the cost of both, performing some measures of execution time, as well as comparing these results with the optimum cost obtained with the IBM ILOG CPLEX optimizer.

**Keywords**— Mission Planning, Manufacturing Systems.

## 1 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 (Service-drone, 2014). 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 were performed autonomously, it would relieve the work of these operators, since they perform tedious and repetitive tasks (Pascarella et al., 2013).

A probable improvement of these logistics systems is the increase of the automation of the UAVs, what results in minimization of the costs, e.g. in terms of time. Consequently, investments and studies related to stand-alone UAVs are important to the development of smart factories (Hern, 2014). However, one of the main problems with the use of autonomous UAVs is the reliability and intelligence of the system. Thus, increased employment of autonomous UAVs requires the development of devices that are capable of performing tasks and interacting with the environment intelligently and reliably.

Autonomous UAVs need to know what will happen in a future instant and what is the best decision to make at the present time, and therefore 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 capabilities of autonomous UAVs (Finn and Scheding, 2012). As a consequence, in order to perform a mission successfully, it is recommended to make a plan to the task (Garecht, 2010). Mission planning problems consist of planning events to meet certain requirements associated with the plan and improving mission objectives (Krozel, 1988). Therefore, this is one of the main challenges faced in solving this type of problem.

Some researches about evaluation and optimization of mission planning have been done in the last years. Schwarz and Sauer (2012) have used ant colony to optimize missions for an automated guided vehicle (AVG). Another paper investigates energy consumption for a factory and evaluate the logistic planning processes using a metric of evaluation(Müller et al., 2012).

This paper presents a methodology that evalu-

ate the cost of mission planners for a commercial UAV. We developed a evaluation metric that evaluate the relative cost of a planning strategy related with the optimal cost generated by the CPLEX optimizer.

Summarizing, the main contributions of this work are:

- planner evaluation metric;

- use of a commercial UAV in intralogistics.

The remaining of this work is organized as follows: Section 2 presents some previous works related to mission planning, optimization and evaluation, Section 3 provides the fundamentals of mission planning and optimization problems, Section 4 explains the proposed evaluation methodology in details, Section 5 shows the experimental procedures and results in order to explore and demonstrate the potential of methodology, and, finally, Section 6 concludes the work.

## 2  Related works

In the literature, there are some attempts to implement UAV guidance systems that perform mission planning. Doherty et al. (2009) presented an architecture of a framework for mission planning and execution tracking applied to an unmanned helicopter. During the execution of the mission, knowledge was acquired through sensors which was used to create state structures. These structures will allow the construction of a logical model, representing the real development of the system and its environment over time. Then, 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 (Whalley et al., 2005), which generates mission plans using a theoretical approach to decision making. A high-level standalone control is provided by the framework Apex (Baer-Riedhart, 1998), 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 was integrated into a robotic helicopter and tested in more than 240 scenarios.

A similar project, called Ressac (Research and Rescue by Cooperative Autonomous System), was conducted by the French Aerospace Laboratory (ONERA) for a search and rescue scenario (Fabiani et al., 2007). This architecture for an exploration mission was developed based on the idea of decomposing the mission into a sequence of tasks or macro-actions associated with rewards. The problem was modeled using a Markov decision process framework (MDP) and dynamic programming algorithms for mission planning. Konigsbuch (Teichteil-Königsbuch and Fabiani, 2007) extends the guidance system and integrates with a robotic helicopter.

Finally, the German Aerospace Center (DLR) has also developed a mission management system based on the behavior paradigm (Adolf and Andert, 2010), which has been integrated with the ARTIS helicopter and validated in different scenarios, including follower of waypoints and search and tracking mission.

## 3  Background

### 3.1  Mission Planning

Firstly, a mission can be defined as a goal that need to be completed. In the context of this work, the mission of the UAV is delivery of packages 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 (loading/dropping a load, taking videos/pictures, etc.), typically over a time period (Ramirez-Atencia et al., 2014). An important term in this work is the concept of planner which is the agent (software implementation) that generate a mission. Functionally, mission planning lies above the trajectory planning process, where the mission planner generates a desired mission plan, and then the trajectory planner generates the flight plan (trajectories) between the waypoints.

### 3.2  Optimization Problems

An optimization problem is about finding the best solution (relative to a certain criterion) among a set of available alternatives. For example, 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.

Two distinctions need to be made to better understand the universe of optimization problems. The first is to distinguish a problem which refers to a more general class, for example, the problem of packaging, and an example representing a special type of problem, for example the problem of packaging, wherein there are 5 packages for packaging 25 objects of different sizes. The second distinction concerns the existence of two categories of problem classes: the abstract problem classes and the concrete problem classes. As the name itself suggests, the second category refers to problems that have "concrete existence," that is, problems

for which instances can be created. The BPP corresponds to this category. Together they are also part of a more abstract class: grouping problems. Only with the problem of abstract classes, it is impossible to define instances. In fact, as shown in Figure 1, the classes of concrete and abstract problems form a hierarchy of optimization problems.

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 (Francq, n.d.).

In order for the optimization problems to be solved, 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 an 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.

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 Be 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 satisfy the set of constraints imposed to be accepted. The formatting of an optimization problem mathematically occurs as follows:

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

Where $A$ is the set of workable solutions that the function $f$ can generate.

One can interpret the space of solutions as being a subset of Euclidean space $\mathbb{R}^n$. Each variable is a dimension of space. For a function with two variables it is possible to form in a two-dimensional space and with the addition of a third dimension to the result of the function with $x$ and $y$ as input it is possible to observe the behaviour of the function as The $x$ and the $y$ of the function undergo variation. In Figure 3 a two-dimensional graph is shown in which the variation of the function value causes changes in the gray scale. For larger values ??a darker gray is obtained and for smaller values ??a lighter gray is obtained. With this, one can observe the space of solutions in a panoramic way in the two-dimensional space and it is verified that as it approaches the center the function generates smaller values. In it it is also possible to notice that the x-shaped marks, which represent the solutions found, vary towards the local minimum that is in the middle of the search space. The heuristic used to search for the optimal solution in this case was to look for neighbouring solutions that minimized the cost of the function in the same way that a sphere that rolls over an inclined plane stabilizes when it reaches a valley that would be the local minimum Function. However, this heuristic is not always the most adequate, as will be seen later.
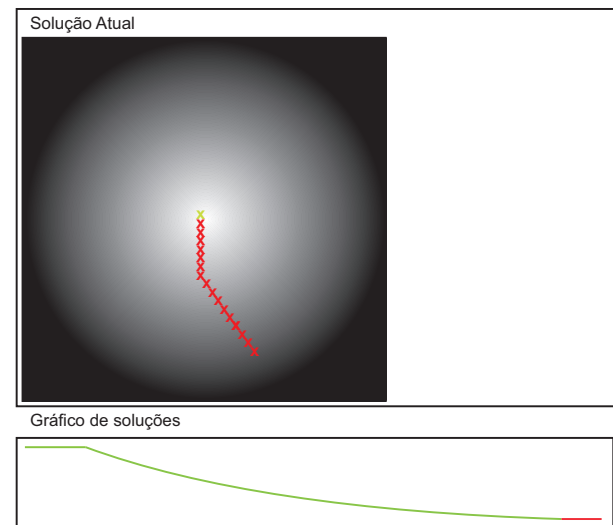


Figure 2: Walking through the space of solutions

# 4 Methodology of Time Cost Evaluation, UAV use and Mission Planning

[IB: Modifique o título (expanda). Metodologia de quê?? Tente fazer algo parecido com o do trabalho.]

In this section, there will have a brief clarification of the contents of this paper.

## 4.1 UAV Movement System

In this section, we first investigate the platform used in this work, verifying the hardware characteristics and how to control it by software. The core hardware of the UAV IRIS+ is the Pixhawk and we can control it using a Python library - dronekit that uses MAVLink[1] protocol. Secondly, we integrate a robot gripper to the UAV to take and leave packages during the mission. After that, we created movement functions to make the UAV

---

[1]MAVLink or Micro Air Vehicle Link is a protocol for communicating with small unmanned vehicle. It is designed as a header-only message marshalling library
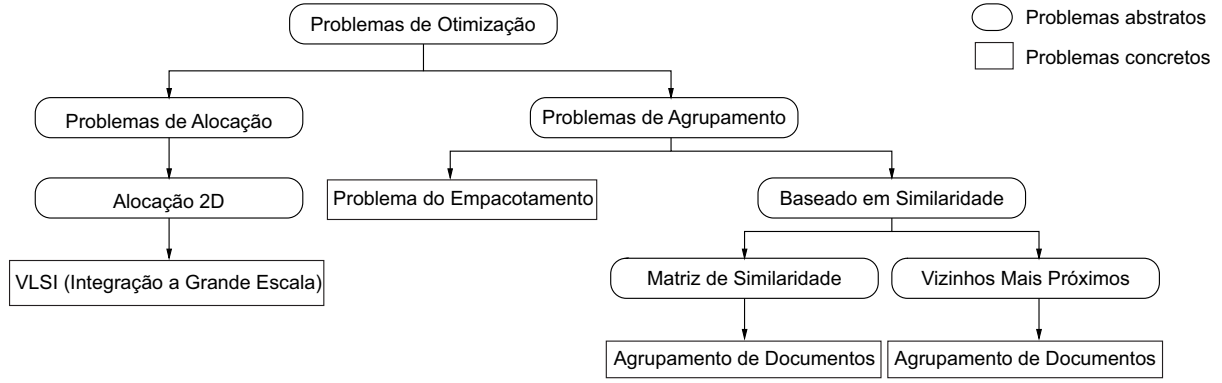
Figure 1: Some Optimization Problems.[IB: Falta apenas traduzir a imagem.]

movement and control it using some resources of the dronekit API.
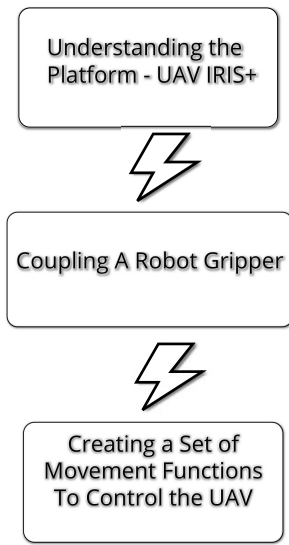


Figure 3: UAV Movement Systems.

### 4.2 Optimization Problem Modelling and Planner Evaluation Technique

The purpose of this subsection is to elaborate a modelling of the mission planning problem in an optimization problem. In order to find later, the shortest execution time of all the tasks (minimization), based on the data below:

It is assumed that the processing time $p_i$ is the total time of production, that is, considering the sum of the time that the UAV stays in the stock collecting inputs, the time in which it stays in the production line leaving inputs and collecting products, And the time it takes for the UAV to move from the production line to the stock or to the customer. We also assume that the production time (setup) $s_i$ is the time the product takes to be produced, after having in line all the inputs needed to produce it. Therefore, the total time for the production of a given product is the sum of the processing time $p_i$ and of setup $s_i$.

In this modeling, the fictive task or dummy of the scheduling, $j_0$, with processing time proportional to the time that the UAV collect the input and leads to the production line. In the literature, production lines are called unrelated parallel machines, for optimization problems of scalable tasks (when there are $m$ machines in parallel with performances depending on the task to be executed, notation used is $R$) (Du and Brucker, 2008).

- $J = \{j_1, j_2, ..., j_n\}$ is the set of tasks;

- $M = \{m_1, m_2, ..., m_n\}$ is the set of machines (production lines);

- $P = \{p_1, p_2, ..., p_n\}$ is the processing time of each task;

- $S = \{s_1, s_2, ..., s_n\}$ is the setup time (production time of each production line);

- $R||C_{max}$ (Three-field notation[2] used in the literature to represent a given task scheduling problem, in this case, minimizing the makespan [3] in an unrelated parallel machine environment (Graham et al., 1979).

Model:

1. **Choosing the constraints:**

   The variable $x_{ij}$ is a binary constraint which takes the value 1 if the task $j$ is running on the machine $i$, and 0 otherwise. The variable $C_{max}$ is the variable that will be optimized.

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

   $$C_{max} = makespan \text{ (duration of a mission)}$$
   (3)

---

[2]Notation $\alpha|\beta|\gamma$, where $\alpha$: processing environment, $\beta$: problem constraints and $\gamma$: optimization criterion.
[3]Makespan is the termination time of the most loaded processor.

2. **Elaboration of the objective function:**
The objective function is the variable $C_{max}$ (total process execution time) that needs to be minimized.

$$\text{Minimize } C_{max} \qquad (4)$$

$C_{max}$ is the variable that need to be minimized (optimized).

3. **Restrictions:**

   • **Each task must be executed/processed in an unique machine (production line)**

$$\sum_{\substack{i \in M \\ \forall j \in J}} x_{ij} = 1 \qquad (5)$$

   For a better understanding what the summation means, let's suppose that there are two production lines and tree products to be produced (tasks), therefore, $x_{00}+x_{10}+x_{20} = 1$, $x_{01}+x_{11}+x_{21} = 1$ and $x_{02} + x_{12} + x_{22} = 1$, so, it can be verified that those restrictions are set, there will be only one task running in a machine.

   • **Time execution of the tasks in each machine**

$$\sum_{\substack{j \in J \\ \forall i \in M}} (p_j + s_j)x_{ij} \leq C_{max} \qquad (6)$$

   For a better understanding what the summation means, let's suppose that there are two machines and two tasks, and the time of processing and setup of each task are $p_0 = 100$, $p_1 = 100$ and $s_0 = 12$, $s_1 = 18$ respectively. Therefore, the production time of the product 0 in the production line 0 is $p_0 + s_0$ and so on.

4. **Completeness and non-negativity:**

$$x_{ij} \in \{0, 1\} \; \forall i = 1, 2, ..., M \text{ and} \\ \forall j = 0(dummy), 1, 2, 3, ..., N \qquad (7)$$

5. **Full model:**

$$\begin{aligned} \text{Minimize} \quad & C_{max} \\ \text{Subject to} \quad & \sum_{\substack{i \in M \\ \forall j \in J}} x_{ij} = 1 \\ & \sum_{\substack{j \in J \\ \forall i \in M}} (p_j + s_j)x_{ij} \leq C_{max} \\ & x_{ij} \in \{0, 1\} \; \forall i = 1, 2, ..., M \; e \; \forall \\ & j = 0(dummy), 1, 2, 3, ..., N \end{aligned}$$

In order to evaluate the cost of the two solutions implemented in this work, a generalized evaluation metric was developed, as shown below.

The cost function evaluated in this work is the time spent for the execution of the mission. The evaluation metric compares the cost of the strategies developed with the cost of a solution generated by a solver.

Firstly, the optimal cost of the problem is obtained through the CPLEX solver, which returns the optimum value (minimum mission execution time). The implementation of the modelling optimization problem is done in C ++ with the help of the CPLEX solver.

Considering,

   • $c_o$ - it is the optimal cost obtained by the CPLEX solver;

   • $c_X$ - it is the cost of the solution generated by planner X;

   • $Crel_X$ - it is the relative cost of the solution generated by planner X.

The evaluation of each mission planner will be made relative to the optimal cost, therefore:

$$Crel_X = \frac{c_o}{c_X} \qquad (8)$$

Where $0 \leq Crel_X \leq 1$, it can be verified that as close of 1 the $Crel_X$ is, the solution cost is smaller.

*4.3 Mission Planners Cost Evaluation Metrics*

To evaluate the cost of mission planner solutions implemented, a generalized evaluation metric was developed, as shown below.

The evaluation function in this work is the execution time of the mission. The evaluation metric compares the cost of the strategies developed with the cost of a solution generated by a solver.

Firstly, the optimal cost of the problem is obtained through the CPLEX solver, which returns the optimal value (minimum mission execution time) under the same circumstances. The implementation of the optimization problem modeled in the subsection 4.3 is implemented in C++ using the CPLEX solver.

Knowing that:

   • $c_o$ - it's the optimal cost obtained by the CPLEX solver;

   • $c_X$ - it's the cost of the solution generated by planner X;

   • $Crel_X$ - it's the relative cost of the solution generated by planner X.

The evaluation of each planner will be made relative to the optimal cost, therefore:

$$Crel_X = \frac{c_o}{c_X} \qquad (9)$$

As we know $0 \leq Crel_X \leq 1$, we can verify that the closer to 1 the $Crel_X$ is, the less the cost of the solution.

## 5 Experimental Evaluation

### 5.1 Case Study

To perform the analysis of the techniques used and to evaluate the cost of the solutions addressed, the following case study was used:
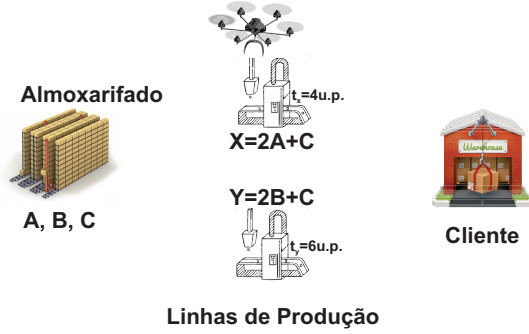


Figure 4: Case Study Representation.

In the Figure 4, we can see that there are inputs in the warehouse of type A, B and C and two production lines that produce products of type X and Y.

Each production line produces only one type of product and has a characteristic production time, as shown in Figure 4. In this case, to produce a product of type X, two inputs of type A and one input of type C are required. In order 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 $(1u.p.)$ was considered to be a `GoTo` command performed by the UAV.

The task to be performed is the production of the customer order, where the UAV will collect supplies from the warehouse, take to the production line and once the production of a certain product is finished, it will lead to the customer.

### 5.2 Mission Planners

In the context of this work, mission planner is a software developed with the purpose of generating a production mission given the warehouse and customer request. This program generates a `.mission` extension file containing a set of mission commands, as shown in the 4.2 subsection.

### 5.2.1 Planner A

In this strategy, the UAV starts moving toward the warehouse, takes two inputs of type A to produce a product X, then takes the type C input, waits on the production line for the product X to be produced. After being produced, it leads to the customer. If there is more products of type X to be produced, it goes to the warehouse and carries out the whole process again.

After all the X products are produced and taken to the customer, the UAV will go to the warehouse and take the two type B inputs and then the type C input, wait for the first product Y to be produced and then take it to the customer. If you need to produce more Y products, the process is the same.

The pseudo code of this algorithm is shown in the Algorithm 1:

---
**Algorithm 1** Planner A
---
**Input**: warehouse
**Input**: order
**Output**: mission file *.mission*
**begin**
    check the order;
    **repeat**
        go to the warehouse
    **until** *production of all X elements finish*;
    ;
    **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;
    **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**

---

### 5.2.2 Planner B

In planner B, unlike planner A where the UAV is idle waiting for each product to be produced and only then takes the customer, the UAV continues the production process while the output of the products does not end. Once the production of each product is finished, the VANT for the task that was running and goes to the production line of that particular product, performs the collection and takes it to the customer. It then returns the task it was previously running, performing a scheduling of tasks.

The pseudo code of this algorithm is shown in the Algorithm 2:

---

**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**
_____

### 5.3 Evaluation Cost

After the implementation of the algorithm of planners A and B, the metric shown in this work was used to evaluate the cost of the algorithms. The results of the proposed mission planning methods in this work were compared to the optimal solution obtained with the branch-and-cut algorithm of the IBM/ILOG CPLEX 12.4 tool developed in C++.

In order to obtain better results in the comparison, it was considered only the time in which the UAV takes to finish the production of a product, excluding the time in which it leaves its initial position and moves to the warehouse, the time in which it Moves from the customer to the warehouse and the time it takes to go to the end point and land. This was done due to the greater practicality of implementing the problem in the CPLEX solver. The problem was modelled considering the total processing time, that is, only the processing time of the task (production) and the time of set-up (time of production of the line). Thus, the execution time of each mission was measured and compared to the value obtained by the CPLEX solver (optimal time), as shown in Table 1.

|  | Time (s) |
|---|---|
| **Planner A** | 420 |
| **Planner B** | 404 |
| **CPLEX** | 134 |

Table 1: Time execution for comparison with the optimal solution time

The Table 1 shows the mission execution times obtained using the scheduler algorithm A, the scheduler algorithm B, and the minimum value given by the solver.

Using the metric shown in the section 4.4, then:

$$Creal_A = \frac{134}{420} = 0,319 \tag{10}$$

$$Creal_B = \frac{134}{408} = 0,328 \tag{11}$$

It can be seen in the Equation 11 that planner B performs the mission more quickly and has a lower cost than planner A, according to the metric used.

### 5.4 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 measurement was performed using the two mission planning algorithms developed, using the case study shown in 5.1.

Below, we can verify the two mission files generated by the two strategies developed in this work:

The following is data about the mission being executed in the simulator SITL [4]. The table below shows the performances of both schedulers relative to flight time.

| | Planner A | Planner B |
|---|---|---|
| Testes | Flight Time (s) | Flight Time (s) |
| 1 | 460,405 | 430,830 |
| 2 | 460,693 | 436,885 |
| 3 | 462,080 | 441,681 |
| 4 | 457,719 | 441,277 |
| 5 | 461,227 | 451,865 |

Table 2: Mission Planners Flight Time - Simulator.

In the Table 2, it can be verified that the mission time executed by planner B presented less time compared to planner A in all five tests.

In Figure 5 is shown a graph of the flight times of both planners in the five tests done in the simulation environment.
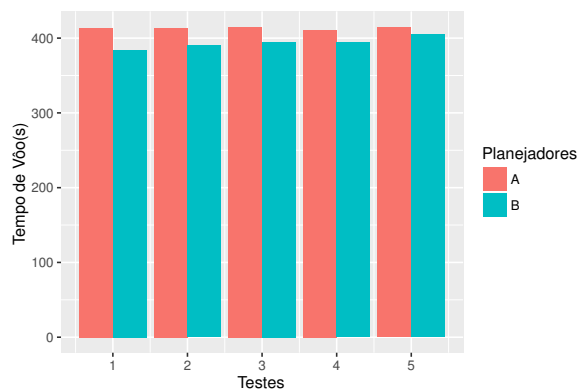


Figure 5: Flight Time Graph of Mission Planners Relative to 5 Tests in Simulator.

Next, you can see the results for the tests performed in real environment used by both planners.

In Figure 6 the map where the tests were performed is shown on the map at the Faculty of Physical Education and Physiotherapy of Federal University of Amazonas.

Figure 6: Warehouse, Production Line X, Production Line Y and Costumer in the Map.

In Table 3, it can be seen that the mission time executed by planner B in real environment also showed a shorter time compared to planner A in the five tests:

| | Planner A | Planner B |
|---|---|---|
| Testes | Flight Time (s) | Flight Time (s) |
| 1 | 455,12 | 441,72 |
| 2 | 456,93 | 440,18 |
| 3 | 457,19 | 447,51 |
| 4 | 460,25 | 438,19 |
| 5 | 459.47 | 445,85 |

Table 3: Mission Planners Flight Time - Real.

In Figure 7 is shown a graph of the flight times of both planners in the five tests done in real environment.
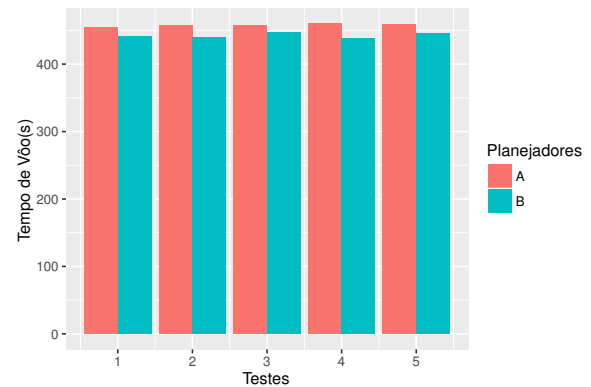


Figure 7: Flight Time Graph of Mission Planners Relative to 5 Tests in a Real Environment.

As shown in the section 5.3, planner B was faster than planner A. This can also be verified in real environment, as shown in tables 2 and 3 of mission execution time in simulated and real environment.

## 6 Conclusions

In this work, a common application of intra logistics performed by a VANT was developed, as

well as the implementation of two mission planning strategies, as well as the cost evaluation of the same, compared to the optimal cost obtained by the CPLEX optimizer.

It is understood that some contributions were made with the conclusion of this work, such as:

In the manipulation and control of a commercial UAV, the ways of manipulating and controlling the 3DR IRIS + UAV of the UFAM Robotics Laboratory were investigated for future uses in undergraduate and postgraduate university projects.

Another important contribution was the creation of a methodology for evaluating the cost of mission planners, since it allows to verify their effectiveness against optimal values obtained by an optimizer.

Finally, it was possible to say that the manipulation and control of the 3DR IRIS+ UAV, as well as the creation of the cost evaluation methodology were successfully applied in simulated and real environments, thus validating the theory shown in the work.

Further works includes the use of computational vision for the recognition of inputs, improve the modeling of the optimization problem for better results in cost evaluation, and use a solver to directly generate an optimized mission.

## References

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

Baer-Riedhart, J. (1998). Nasa environmental research aircraft and sensor technology program, *Aerospace America, January* .

Doherty, P., Kvarnström, J. and Heintz, F. (2009). A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems, *Autonomous Agents and Multi-Agent Systems* **19**(3): 332–377.

Du, D. and Brucker, P. (2008). Scheduling algorithms.

Fabiani, P., Fuertes, V., Piquereau, A., Mampey, R. and Teichteil-Königsbuch, F. (2007). Autonomous flight and navigation of vtol uavs: from autonomy demonstrations to out-of-sight flights, *Aerospace Science and Technology* **11**(2): 183–193.

Finn, A. and Scheding, S. (2012). *Developments and challenges for autonomous unmanned vehicles*, Springer.

Francq, P. (n.d.). Optimization problems.
**URL:** *http://www.otlet-institute. org/wikics/Optimization_Problems. html*

Garecht, J. (2010). How to write a successful fundraising plan. Accessed in: 17/10/2015.
**URL:** *http://www. thefundraisingauthority.com/ fundraising-basics/fundraising-plan/*

Graham, R. L., Lawler, E. L., Lenstra, J. K. and Kan, A. R. (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey, *Annals of discrete mathematics* **5**: 287–326.

Hern, A. (2014). Dhl launches first commercial drone 'parcelcopter'delivery service, *The Guardian* .

Krozel, J. A. (1988). Search problems in mission planning and navigation of autonomous aircraft, *Technical report.*

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

Pascarella, D., Venticinque, S. and Aversa, R. (2013). Autonomic agents for real time uav mission planning, *Ubiquitous Intelligence and Computing, 2013 IEEE 10th International Conference on and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC)*, IEEE, pp. 410–415.

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

Schwarz, C. and Sauer, J. (2012). Towards decentralised agv control with negotiations., *STAIRS*, pp. 270–281.

Service-drone (2014). Drones for logistic and transport | multirotor. Accessed in: 12/02/2016.
**URL:** *http://www.service-drone. com/en/production/ logistics-and-transport*

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

Whalley, M., Freed, M., Harris, R., Takahashi, M., Schulein, G. and Howlett, J. (2005). Design, integration, and flight test results for

an autonomous surveillance helicopter, *Proceedings of the AHS International Specialists' Meeting on Unmanned Rotorcraft*.