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

Use of a commercial UAV - 3DR IRIS + - in the intra-logistics [1], aiming to give another option of agility in the manufacturing process. Additionally, it will be approached a study of evaluation of the cost of the missions that the UAV executes in this process.

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 UAVs, which are any and every type of aircraft that does not require on board pilots to be guided. 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. However, one of the main problems with the use of autonomous UAVs is the reliability and intelligence of the system. Because of these factors, investments and studies related to stand-alone UAVs are still considered small (Hern, 2014). 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 re-

[1] Internal Logistics of movement and storage.

quirements 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.

This paper presents a methodology that evaluate the cost of mission planners for a commercial UAV. The evaluation is made by comparing the mission execution time with the cost of an optimal solution solved by CPLEX solver. Additionally, a middleware was developed to interface the mission planning application and the embedded control software, adapting the UAV for intralogistics. Finally, experiments were done to verify the consistency of evaluation methodology.

The remaining of this work is organized as follows: A background section was inserted to give a better understanding of mission planning and optimization problems. There is methodology section to show the explanation of the methods that this project uses. The next section shows the experimental results of this work. Finally, last section presents the conclusion of this project.

2 Background

2.1 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.

2.1.1 Related Works in Mission Planning

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.

2.2 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:

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

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 2 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.

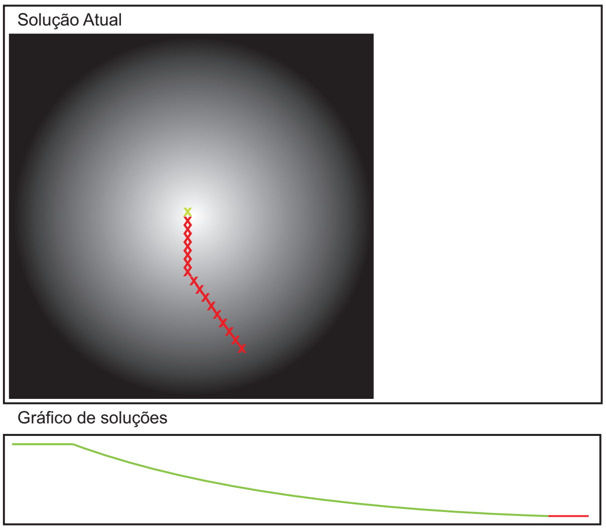

Figure 2: Walking through the space of solutions

3 Methodology

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

3.1 UAV Movement System

3.2 Mission Planning

3.3 Optimization Problem Modeling and Planner Evaluation Technique

The purpose of this subsection is to elaborate a modeling 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

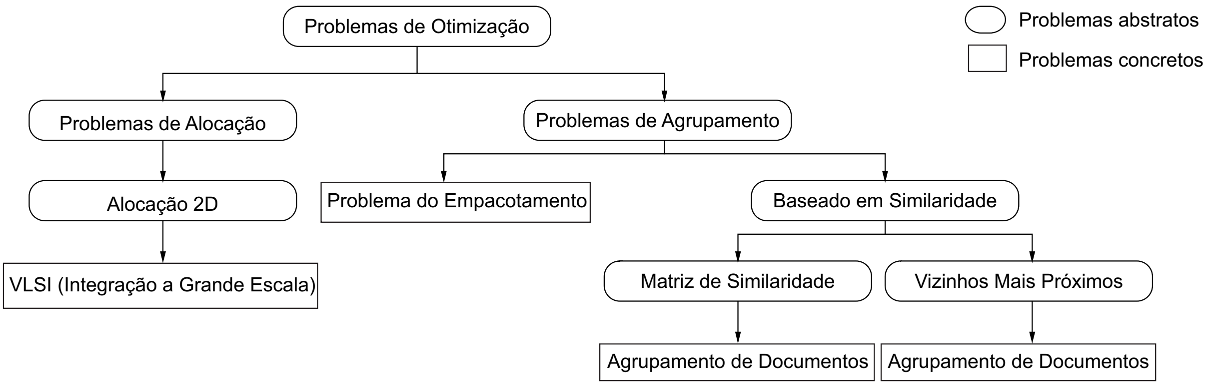

Figure 1: Some Optimization Problems.[IB: Falta apenas traduzir a imagem.]

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.

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

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

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

$$\text{Minimize } C_{max} \tag{4}$$

   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)**

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

     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**

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

     For a better understanding what the summation means, let's suppose that there are two machines and two tasks,

[2] Notation $\alpha|\beta|\gamma$, where α: processing environment, β: problem constraints and γ: optimization criterion.

[3] Makespan is the termination time of the most loaded processor.

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:**

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

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:

$$Crel_X = \frac{c_o}{c_X} \tag{8}$$

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.

3.4 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 3.3 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:

$$Crel_X = \frac{c_o}{c_X} \tag{9}$$

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.

4 Experimental Evaluation

4.1 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:

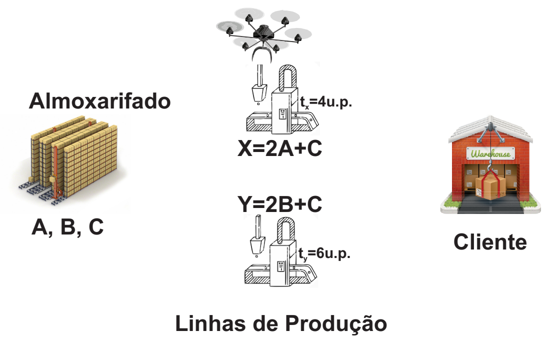

Figure 3: Case Study Representation.

In the Figure 3, 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 3. 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.

4.2 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 3.2 subsection.

4.2.1 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
```

4.2.2 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
```

4.3 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 3.4, 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.

4.4 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 4.1.

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

[4]Simulator that allows executing a Plane, Copter or Rover without the need of a hardware)

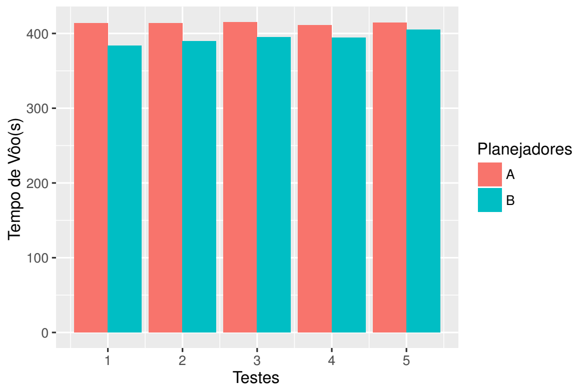

Figure 4: Flight Time Graph of Mission Planners Relative to 5 Tests in Simulator.

Next, you can see the results for the tests performed in real environment used by both planners. In Figure 5 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 5: 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 6 is shown a graph of the flight times of both planners in the five tests done in real environment.

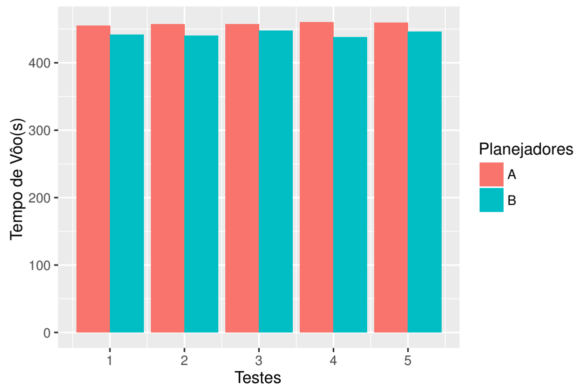

Figure 6: Flight Time Graph of Mission Planners Relative to 5 Tests in a Real Environment.

As shown in the section 4.3, 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.

5 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.

6 Como utilizar o estilo SBATEX

Este texto serve de guia de utilização do estilo SBATEXpara LATEX 2_ε. Com este estilo é possível gerar artigo no formato utilizado pelo *Congresso Brasileiro de Automática* e *Simpósio Brasileiro de*

Automação Inteligente na produção das versões acabadas dos artigos nos anais.
O seguinte parâmetro deve ser passado (entre colchetes) como argumento do comando `\documentclass`, que deve ser o primeiro comando a aparecer no artigo. Por exemplo, o comando

```
\documentclass[conference]{sbatex}
```

é utilizado no início deste arquivo para geração do formato final para o *Congresso Brasileiro de Automática* e o *Simpósio Brasileiro de Automação Inteligente*.
Pode-se adicionar ao comando `\documentclass` a seguinte opção:

- `harvard`: inclui o pacote `harvard`, utilizado para produção das referências bibliográficas. Pode ser usado em conjunto com qualquer das outras opções. Veja seção 6.4.

Por exemplo, para propor um artigo ao congresso e utilizar o pacote de referências bibliográficas `harvard` usa-se o comando

```
\documentclass[conference,harvard]{sbatex}
```

A ordem dos parâmetros não é relevante.
Os seguintes *pacotes* LaTeX estão disponíveis automaticamente[5] com o estilo SBATeX:

- `ifthen`: pacote com comandos de controle de fluxo do tipo ‘if’, ‘then’ e ‘else’; utilizado internamente para controle das opções de formatação.
- `fancyhdr`: pacote para formatação dos cabeçalhos e rodapés.
- `babel`: pacote para escrita em múltiplos idiomas; os idiomas/dialetos `english`, `brazil` e `spanish` já estão habilitados.
- `theorem`: pacote para controle dos ambientes `theorem`, `lemma`, `corollary` e `proof`.
- `geometry`: pacote para controle da dimensões do texto na página.
- `latexsym`: pacote com símbolos extras do LaTeX 2_ε.

A opção `submission` provê o pacote:

- `setspace`: pacote para controle de espaçamento duplo entre linhas.

E a opção `harvard` provê o pacote:

- `harvard`: pacote para formatação das referências bibliográficas.

Não é necessário incluir explicitamente as cláusulas `\usepackage` correspondentes a estes pacotes no artigo para que se possa utilizá-los. Caso a instalação do LaTeX não contenha alguns destes pacotes, consulte a página

[5] Isto também significa que *é necessário ter estes pacotes instalados* para que se possa utilizar o estilo SBATeX.

```
http://www.ctan.org
```

Com a opção `conference`, e devido à escolha de um conjunto de fontes diferentes, recomenda-se subsituir o pacote `fontenc` pelo pacote `ae` (*Almost European Computer Modern*), o que pode ser feito com o comando

```
\usepackage{ae}
```

Neste caso não é necessário adicionar explicitamente o pacote `fontenc`, que já é carregado automaticamente pelo pacote `ae`. Veja também a seção 6.5.
Após o preâmbulo, inicia-se o texto propriamente dito por meio do comando

```
\begin{document}
```

O passo seguinte é a declaração do título do artigo, autores e afiliações.

6.1 *Título, autores e afiliações*

O título do artigo é declarado por meio do comando `\title`. Por exemplo, o título deste artigo foi gerado com o comando:

```
\title{Artigo Exemplo}
```

O título não deve conter agradecimentos, os quais devem aparecer em uma seção separada, no final do artigo.
Em seguida, definem-se os autores com seus endereços eletrônicos e afiliações por meio dos comandos `\author` e `\address`. Por exemplo, o nome e endereço do primeiro autor deste artigo, João da Silva, é introduzido por meio dos comandos:

```
\author{João da Silva}
       {jsilva@silva.com}
\address{Endereço do João e da Maria\\
         Em algum lugar\\
         Cidade, Estado, País}
```

Note-se que o comando `\author` requer dois argumentos: o nome e o endereço eletrônico. Quebras de linha podem ser obtidas na afiliação por meio do comando `\\`. Segue-se, neste exemplo, a declaração do nome e afiliação do segundo autor:

```
\author{Joaquim Pereira}
       {jpereira@pereira.org}
\address{Endereço do Joaquim\\
         Em algum lugar\\
         Cidade, Estado, País}
```

Em geral, para cada comando `\author` deve haver um comando `\address` correspondente, salvo o caso de compartilhamento de endereços. Neste caso, omite-se o comando `\address` para o segundo autor e indica-se o número do endereço que se deseja compartilhar como um parâmetro adicional do comando `\author`. Por exemplo, o comando:

```
\author[1]{Maria da Silva}
          {maria@pereira.org}
```

indica que a Maria da Silva utilizará o endereço de número 1, que corresponde ao endereço de João da Silva.

Os autores seguintes possuem endereços independentes e são introduzidos da maneira convencional:

```
\author{Pedro Manoel}
       {pedro@pedro.com}
\address{Endereço do Pedro e do Rafael\\
         Em algum lugar\\
         Cidade, Estado, País}

\author{José Rodrigues}
       {jose@rodrigues}
\address{Endereço do José\\
         Em algum lugar\\
         Cidade, Estado, País}
```

Finalmente, o autor Rafael é definido como

```
\author[3]{Rafael Pires}
          {rafael@pires}
```

o que indica que Rafael compartilha o endereço de número 3 com Pedro Manoel. Lembre-se de que, apesar de Pedro Manoel ser o autor de número 4, o seu endereço é o de número 3, uma vez que a Maria já compartilha – sem segundas intenções – do mesmo endereço com o João.
O comando

```
\maketitle
```

coloca o título e os autores no formato adequado.

6.2 Resumo, abstract e palavras-chave

Os artigos devem conter um resumo em português (ou espanhol) e em inglês. O ambiente para definição dos resumos, em qualquer idioma, é o `abstract`. Para diferenciar os idiomas utiliza-se o comando `\selectlanguage`, do pacote `babel`. Por exemplo, a seqüência de comandos produz um *Abstract* seguido de um *Resumo*:

```
\selectlanguage{english}
\begin{abstract}
  Yossarian says, ...
\end{abstract}

\selectlanguage{brazil}
\begin{abstract}
  O Largo da Sé ...
\end{abstract}
```

As *palavras-chave* devem ser definidas por meio do comando:

```
\keywords{Exemplo, Ilustração}
```

Deste ponto em diante selecione o idioma a ser utilizado e inicie o texto da sua contribuição.
IMPORTANTE: Caso esteja se produzindo um texto com a opção `conference`, colocam-se os comandos que vão de `\maketitle` a `\keywords` dentro de um ambiente `twocolumn`. Isto é necessário para que os resumos sejam formatados em coluna simples[6]. Veja seção 6.5.

[6]Como este arquivo está adaptado para utilizar o formato adequado para o *Congresso Brasileiro de Automática* e o *Simpósio Brasileiro de Automação Inteligente*, isto já está feito

6.3 Teoremas, lemas, corolários e provas

Quatro ambientes já se encontram pré-definidos no estilo SBATEX. São eles

- `theorem`: ambiente para teoremas.
- `corollary`: ambiente para corolários.
- `lemma`: ambiente para lemas.
- `proof`: ambiente para provas.

Segue-se um exemplo de utilização destes ambientes:

Lema 1 (Desigualdade Subtrativa) *Em um espaço vetorial linear normado* $\|x\|-\|y\| \leq \|x-y\|$ *para quaisquer vetores* x, y.

Prova:

$$\begin{aligned} \|x\| - \|y\| &= \|x - y + y\| - \|y\| \\ &\leq \|x - y\| + \|y\| - \|y\| \\ &= \|x - y\|. \end{aligned}$$

□

Este lema é produzido pela seguinte sequência de comandos:

```
\begin{lemma}[Desigualdade Subtrativa]
  Em um espaço vetorial linear ...
\end{lemma}
\begin{proof}
  ...
\end{proof}
```

Estes ambientes, com exceção do ambiente `proof`, aceitam um parâmetro opcional que define um nome. No exemplo acima, o nome *Desigualdade Subtrativa* foi produzido por meio deste argumento.

6.4 Bibliografia

As referências são reunidas ao fim do manuscrito, arranjadas alfabeticamente pelo primeiro autor e cronologicamente para cada autor.
IMPORTANTE: Todas referências citadas devem aparecer em algum outro ponto do texto.
As citações seguem um estilo autor/ano (o mesmo usado na revista *Automatica*). O pacote `harvard` (já incluído pelo SBATEX) provê vários comandos para inclusão de citações. Dois dois mais utilizados são o tradicional `\cite` e o `\citeasnoun`. Veja um exemplo:

```
O resumo deste artigo é um trecho do
livro~\citeasnoun{serafim}. Já o abstract é
extraído de~\citeasnoun{catch22}.
Informações sobre o estilo bibliográfico
\verb+harvard+ encontram-se
em~\citeasnoun{harvard}. Os interessados
devem consultar referências adicionais
sobre \LaTeX~\cite{latex,%
latex:guide,latex:companion}.
```

produz o trecho:
O resumo deste artigo é um trecho do livro de Andrade (1933). Já o abstract é extraído de Heller (1996). Informações sobre o estilo bibliográfico `harvard` *encontram-se em Williams and Schnier (1998). Os interessados devem consultar referências adicionais sobre LATEX (Lamport, 1986; Kopka and Daly, 1993; Gossens and Mittelbach, 1993).*
Recomenda-se a utilização do programa BibTeX para gerar as suas referências (Lamport, 1986; Kopka and Daly, 1993; Gossens and Mittelbach, 1993). Note-se que não é necessário definir o estilo bibliográfico com o comando `\bibliographystyle`, pois o mesmo é definido automaticamente pelo estilo SBATEX. Basta incluir o arquivo de bibliografias, neste exemplo o artigo `exemplo.bib` por meio do comando.

```
  ...

  \bibliography{exemplo}

\end{document}
```

Note-se que este comando deve ser colocado logo antes do encerramento do artigo, o que é feito pelo comando `\end{document}`, de tal forma a garantir que a bibliografia seja o último item do artigo.

6.5 Esqueleto deste arquivo no formato SBATEX

Este arquivo, adaptado para gerar o formato adequado para o *Congresso Brasileiro de Automática* e o *Simpósio Brasileiro de Automação Inteligente*, faz uso da seguintes instruções:

```
\documentclass[conference,harvard,brazil,english]{sbatex}
  \usepackage[utf8]{inputenc}
  \usepackage{ae}

    ...

    \twocolumn[
      \maketitle

      \selectlanguage{english}
      \begin{abstract}
        Yossarian says, ...
      \end{abstract}

      \selectlanguage{brazil}
      \begin{abstract}
        O Largo da Sé ...
      \end{abstract}

      \keywords{Exemplo, Ilustração}
    ]

    ...
```

7 Contribuição deste artigo

7.1 Os vinte e sete erros mais comuns (de uma lista de cem)

Erros gramaticais e ortográficos devem, por princípio, ser evitados. Alguns, no entanto, como ocorrem com maior frequência, merecem atenção redobrada. O primeiro capítulo deste manual inclui explicações mais completas a respeito de cada um deles. Veja os cem mais comuns do idioma e use esta relação como um roteiro para fugir deles (Filho, 1992).

1. "Mal cheiro", "mau-humorado". Mal opõe-se a bem e mau, a bom. Assim: mau cheiro (bom cheiro), mal-humorado (bem-humorado). Igualmente: mau humor, mal-intencionado, mau jeito, mal-estar.

2. "Fazem" cinco anos. Fazer, quando exprime tempo, é impessoal: Faz cinco anos. / Fazia dois séculos. / Fez 15 dias.

3. "Houveram" muitos acidentes. Haver, como existir, também é invariável: Houve muitos acidentes. / Havia muitas pessoas. / Deve haver muitos casos iguais.

4. "Existe" muitas esperanças. Existir, bastar, faltar, restar e sobrar admitem normalmente o plural: Existem muitas esperanças. / Bastariam dois dias. / Faltavam poucas peças. / Restaram alguns objetos. / Sobravam idéias.

5. Para "mim" fazer. Mim não faz, porque não pode ser sujeito. Assim: Para eu fazer, para eu dizer, para eu trazer.

6. Entre "eu" e você. Depois de preposição, usa-se mim ou ti: Entre mim e você. / Entre eles e ti.

7. "Há" dez anos "atrás". Há e atrás indicam passado na frase. Use apenas há dez anos ou dez anos atrás.

8. "Entrar dentro". O certo: entrar em. Veja outras redundâncias: Sair fora ou para fora, elo de ligação, monopólio exclusivo, já não há mais, ganhar grátis, viúva do falecido.

9. "Venda à prazo". Não existe crase antes de palavra masculina, a menos que esteja subentendida a palavra moda: Salto à (moda de) Luís XV. Nos demais casos: A salvo, a bordo, a pé, a esmo, a cavalo, a caráter.

10. "Porque" você foi? Sempre que estiver clara ou implícita a palavra razão, use por que separado: Por que (razão) você foi? / Não sei por que (razão) ele faltou. / Explique por que razão você se atrasou. Porque é usado nas respostas: Ele se atrasou porque o trânsito estava congestionado.

11. Vai assistir "o" jogo hoje. Assistir como presenciar exige a: Vai assistir ao jogo, à missa, à sessão. Outros verbos com a: A

medida não agradou (desagradou) à população. / Eles obedeceram (desobedeceram) aos avisos. / Aspirava ao cargo de diretor. / Pagou ao amigo. / Respondeu à carta. / Sucedeu ao pai. / Visava aos estudantes.

12. Preferia ir "do que" ficar. Prefere-se sempre uma coisa a outra: Preferia ir a ficar. É preferível segue a mesma norma: É preferível lutar a morrer sem glória.

13. O resultado do jogo, não o abateu. Não se separa com vírgula o sujeito do predicado. Assim: O resultado do jogo não o abateu. Outro erro: O prefeito prometeu, novas denúncias. Não existe o sinal entre o predicado e o complemento: O prefeito prometeu novas denúncias.

14. Não há regra sem "excessão". O certo é exceção. Veja outras grafias erradas e, entre parênteses, a forma correta: "paralizar" (paralisar), "beneficiente" (beneficente), "xuxu" (chuchu), "previlégio" (privilégio), "vultuoso" (vultoso), "cincoenta" (cinqüenta), "zuar" (zoar), "frustado" (frustrado), "calcáreo" (calcário), "advinhar" (adivinhar), "benvindo" (bem-vindo), "ascenção" (ascensão), "pixar" (pichar), "impecilho" (empecilho), "envólucro" (invólucro).

15. Quebrou "o" óculos. Concordância no plural: os óculos, meus óculos. Da mesma forma: Meus parabéns, meus pêsames, seus ciúmes, nossas férias, felizes núpcias.

16. Comprei "ele" para você. Eu, tu, ele, nós, vós e eles não podem ser objeto direto. Assim: Comprei-o para você. Também: Deixe-os sair, mandou-nos entrar, viu-a, mandou-me.

17. Nunca "lhe" vi. Lhe substitui a ele, a eles, a você e a vocês e por isso não pode ser usado com objeto direto: Nunca o vi. / Não o convidei. / A mulher o deixou. / Ela o ama.

18. "Aluga-se" casas. O verbo concorda com o sujeito: Alugam-se casas. / Fazem-se consertos. / É assim que se evitam acidentes. / Compram-se terrenos. / Procuram-se empregados.

19. "Tratam-se" de. O verbo seguido de preposição não varia nesses casos: Trata-se dos melhores profissionais. / Precisa-se de empregados. / Apela-se para todos. / Conta-se com os amigos.

20. Chegou "em" São Paulo. Verbos de movimento exigem a, e não em: Chegou a São Paulo. / Vai amanhã ao cinema. / Levou os filhos ao circo.

21. Atraso implicará "em" punição. Implicar é direto no sentido de acarretar, pressupor: Atraso implicará punição. / Promoção implica responsabilidade.

22. Vive "às custas" do pai. O certo: Vive à custa do pai. Use também em via de, e não "em vias de": Espécie em via de extinção. / Trabalho em via de conclusão.

23. Todos somos "cidadões". O plural de cidadão é cidadãos. Veja outros: caracteres (de caráter), juniores, seniores, escrivães, tabeliães, gângsteres.

24. O ingresso é "gratuíto". A pronúncia correta é gratúito, assim como circúito, intúito e fortúito (o acento não existe e só indica a letra tônica). Da mesma forma: flúido, condôr, recórde, avaro, ibéro, pólipo.

25. A última "seção" de cinema. Seção significa divisão, repartição, e sessão equivale a tempo de uma reunião, função: Seção Eleitoral, Seção de Esportes, seção de brinquedos; sessão de cinema, sessão de pancadas, sessão do Congresso.

26. Vendeu "uma" grama de ouro. Grama, peso, é palavra masculina: um grama de ouro, vitamina C de dois gramas. Femininas, por exemplo, são a agravante, a atenuante, a alface, a cal, etc.

27. "Porisso". Duas palavras, por isso, como de repente e a partir de.

7.2 Observação importante

Os erros de português porventura presentes neste artigo

- ou foram introduzidos propositalmente para deleite dos leitores atentos,
- ou não constam da lista da seção 7.1.

8 Conclusões

Nada resta senão desejar-lhe boa sorte na preparação de seu artigo. Contamos com seu trabalho e sua presença no SBAI 2017 a ser realizado em Porto Alegre-RS.

Agradecimentos

Mencione aqui seus agradecimentos as agências de fomento e colaboradores no trabalho.

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* .

de Andrade, O. (1933). *Serafim Ponte Grande*, Editora Globo.

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.

Filho, E. L. M. (1992). *Manual de Redação e Estilo*, Maltese.

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/*

Gossens, M. and Mittelbach, F. (1993). *The LaTeX Companion*, Addison-Wesley, Reading, MA.

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.

Heller, J. L. (1996). *Catch - 22*, reprint edn, Scribner.

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

Kopka, H. and Daly, P. W. (1993). *A Guide to LaTeX: Document Preparation for Beginners and Advanced Users*, Addison-Wesley.

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

Lamport, L. (1986). *LaTeX: A Document Preparation System*, Addison-Wesley.

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.

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*.

Williams, P. and Schnier, T. (1998). The harvard family of bibliography styles. Documentação que acompanha o pacote `harvard`.