A Study in Search Pattern Efficiency Using Under-Actuated Aircraft

Tiago Carvalho, Leonardo Ferreira, and Daniel Castro Silva

Abstract—**Area coverage algorithms or Search Patterns have always been useful in many real life problems such as search and rescue missions, crime prevention, forest fire detection and many others. This means knowing which algorithms to apply in the different cases in order to obtain the best results is immensely valuable. The goal of this paper is to evaluate mission performance and with it discover the best approach for various possible problems. In order to achieve this goal, multiple metrics were created to have some indicators that can be used to calculate search pattern performance. Various simulations were conducted, with different strategies and parameters, to compare distinct approaches to the various missions. These experiments consisted in applying different search patterns in a search mission with distinct delimitation areas and parameters. The results obtained allow the user to easily adopt the most efficient strategy in the scope of the simulations executed. For example, the larger the ratio between height and width of the delimitation area, the better the Parallel Track pattern is, when compared to the others. On the other hand, the closer the area is to a circle, the better the Spiral search pattern is, when compared to the others.**

Index Terms—**Data collection, search patterns, simulation evaluation.**

I. Introduction

Search patterns consist of paths created with the intent of covering certain areas efficiently. They are useful for various real-world problems such as cleaning robots, search and rescue missions [1], [2], forest fire [3], measurements in an area [4] and many others.

For this reason, researching which search pattern is the most efficient in covering areas of distinct shapes and sizes with different goals is worthwhile, as it would lead to the reduction of resources spent, such as energy and time. However, determining which search pattern yields the best results is not a trivial task, as its efficiency depends on multiple factors, such as area shape and size, existing obstacles, environment and even the characteristics of the vehicle performing the task.

This work is focused on measuring the efficiency of different search patterns, in different scenarios, of under-actuated aircraft. These scenarios may include search missions, forest fire detection or measurements of atmosphere contents, which require coverage of vast areas in the real world. Since both vehicles and fuel are expensive, reducing the resources wasted by choosing the most efficient search pattern is worthwhile.

In order to measure search pattern efficiency, a set of metrics were devised, such as covered or overlapped area, time and fuel spent, and distance covered, among others. The reason for having multiple metrics resides in the fact that the same mission may have different performance depending on user priorities; for instance, the user might prioritize mission time over spent fuel when searching for a missing person or detecting forest fires.

In this work, we implemented three basic search patterns – Parallel Track, Spiral and Expanding Shape – and variations, and experiments were performed in a simulated environment capable of realistic aircraft movement simulation using different areas, vehicles and search radius [5].

The remainder of this paper is structured as follows: in Section II we present some state-of-the-art related works, followed by the methodology and conceived solution in Section III, with the experimental setup in Section IV and results presented in Section V. Finally, Section VI summarizes the conclusions and reflects on limitations of the system, improvements and future work.

II. Related Work

This chapter details some works that contribute to the area by presenting search patterns and/or evaluation metrics that can be used to assess their performance.

Bosse *et al.* evaluate the performance of a coverage algorithm using under-actuated car-like vehicles in a real-life uncertain environment, with dynamic obstacle avoidance [6]. They opted to mix a more dynamic version of the Spiral In search pattern with the Spiral Shift one since it would require fewer sharp turns and would allow for a better degree of freedom to avoid obstacles.

To compare the efficiency of the pattern, they created two experiments, using three metrics (Percentage of Area Covered, Percentage of Area Overlapped, and Time Spent) to evaluate the results:

- Area coverage on a smooth surface (cement floor) without any obstacles, which serves as a baseline to compare with the other one.
- Area coverage on a grassy surface with several trees.

In Fig. 1 (a) we can see the results obtained for the baseline experiment, having obtained 99% of area coverage in 12 minutes, while in Fig. 1 (b) the area covered was 95% in 15 minutes. The area overlapped is presented in both figures with different colors depending on the amount of times said area was covered.

Although containing some similarities to the work presented herein, it does not study the impact the area shape and different vehicles may have in the search pattern

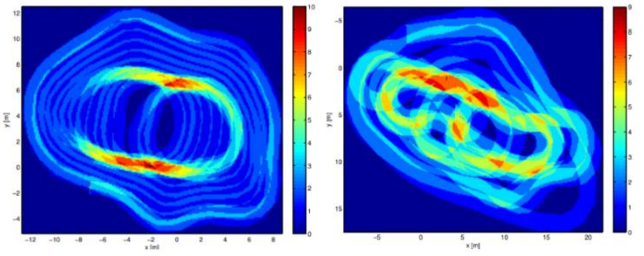
Manuscript received May 18, 2021; revised July 22, 2021.

Tiago Carvalho and Leonardo Ferreira are with the University of Porto, Portugal (e-mail: trvc@fe.up.pt, leonardo.ferreira@fe.up.pt).

Daniel Castro Silva is with the Department of Informatics Engineering, Faculty of Engineering, University of Porto, Portugal (e-mail: dcs@fe.up.pt).

DOI: 10.7763/IJET.2021.V13.1190

performance, nor does it test other search patterns. It is, however, interesting to take into consideration the metrics used.



(a) First experiment (b) Second experiment

Fig. 1: Coverage map of the experiments (extracted from [6]).

Goh and Fan described a mission of seabed depth mapping using a simulator called MOOS-IvP [7] and evaluated different search strategies while doing so [8]. Although they initially considered the search patterns shown in Fig. 2, they rapidly discarded all of them except for the Parallel Track. Such decision was made due to the need of avoiding image overlapping, which would lead to distorted images. It was also required for the vehicle to turn outside of the delimitation area for the same reason.

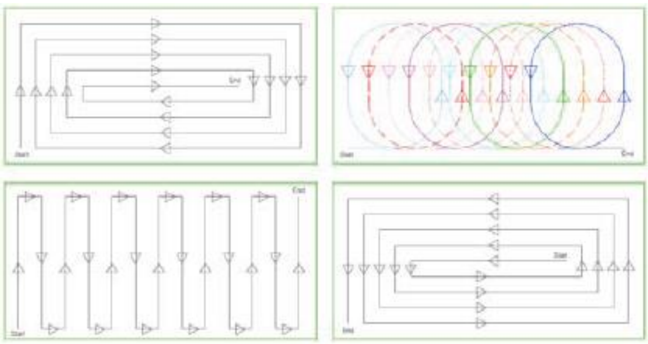


Fig. 2: Search patterns considered in the seabed mapping experiment (extracted from [8]).

The authors were able to map the seabed correctly with little distortion. Most of their study was related to the compensation of water currents, which, albeit not related to the work done, is interesting to learn for possible future disturbances in the simulator, such as strong air currents. This work was important to be aware of the search patterns considered and know more applications of them.

Arbeit tried to apply multiple search patterns in a target search mission, where the sensor could be controlled in two ways, and evaluate their performance [9]. It tries to simulate search missions by using the Parallel Track, the Creeping Line, Expanding Square and Trackline patterns, considering the sensor as a movable camera with a certain field of view. In order to move the sensor, two main approaches were used:

- Pattern-based sensor control: The direction of the sensor is the same as the pattern, with some differences when turning.
- Occupancy Map-based sensor control: The occupancy map has information about the possible position of the target and uses that information to decide the camera orientation in order to maximize the amount of area covered per unit of time. The direction taken by the sensor points to the most probable space of where the target might be; this, however, makes it so the sensor avoids looking at already covered area, since it is known the target is not there.

To evaluate the results, the following metrics were implemented:

- Coverage Factor: A metric that estimates the effectiveness of a given search mission. It is similar to the sensor radius divided by the division between the tracks, but with some differences due to using a directional sensor.
- Search Duration: The duration of the mission.
- Percent Searched: The amount of area searched by the sensor. Calculated by dividing the already searched cells in the occupancy grid by the total number of cells.
- Post-Search Fuel Time Remaining: The amount of time the AUV has before running out of fuel after the completion of the mission.

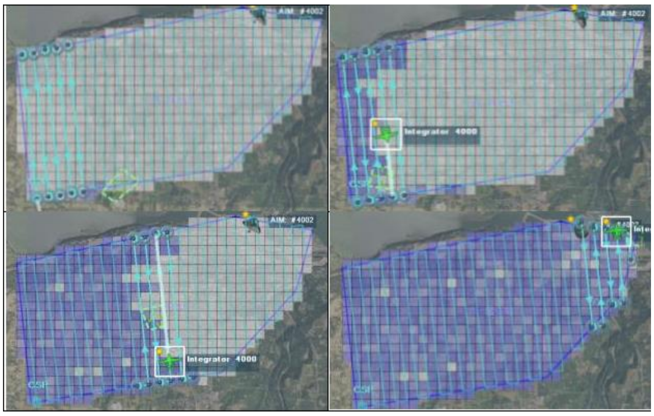


Fig. 3: Creeping line search progression snapshots with the occupancy map-based sensor control algorithm (extracted from [9]).

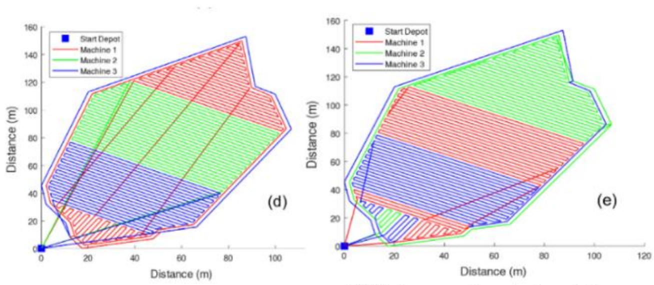
In order to discover if using different sensor controls would change the performance of the mission, multiple experiments were executed using the different search patterns with different sensor controls, such as the one represented in Fig. 3. In the end, Occupancy Map-based sensor control was more efficient in general by a small margin.

However, this thesis does not seem to consider under-actuated vehicles and does not study its effect on the search patterns. It differs from this work in the results that were tried to achieve, which were discovering the efficiency of the sensor control strategies compared to measuring the efficiency of the search patterns and its relation with the mission parameters.

Seyyedhasani *et al.* studied the efficiency of two routing algorithms in a multi-vehicle scenario [10]. They used a mix of the Creeping Line and Parallel Tracking search patterns and separated the area to search between vehicles using a Clarke-Wright heuristic and a tabu search meta-heuristic [11], which are represented in Fig. 4.

Their experiments consisted of using three search patterns (Parallel Track, Creeping Line and a combination of both), one hundred different field shapes and a different number of vehicles in order to compare the two heuristics.

In the end they conclude that the tabu search yielded better results overall but that it took much more time arriving to a solution, while Clarke-Wright would give an acceptable solution much faster.



(a) Clarke-Wright heuristic (b) Tabu search meta-heuristic

Fig. 4: Example of a solution created by both heuristics (extracted from [10]).

III. Methodology and Implementation

This section presents the methodology and describes the simulation platform used, the implementation of the search patterns and their shortcomings, and the metrics used to evaluate the performance.

A. Methodology

In order to achieve the goal of discovering how search strategies are affected by the various parameters of the mission, there is the need to follow the following steps:

- The implementation of the search patterns in a simulator. This simulator should allow for data collection and analysis under different metrics.
- The creation of metrics capable of measuring the patterns performance correctly.
- The execution of multiple missions with different parameters (search area shape, sensor radius, aircraft) and search patterns in order to collect data to analyze.
- Analysis of the collected data, culminating in the discovery of various relations between mission parameters and search pattern performance.

B. Simulation Environment

This work was developed on top of a simulation platform whose main objective is to simulate realistic multi-vehicle missions where vehicles are capable of self-coordination. It makes use of Microsoft Flight Simulator X [12] for realistic simulation, and controls its scenario elements, such as aircraft, via the SimConnect API [13]. This platform allows the user to choose different search patterns to execute the missions, while allowing him to define the aircraft used, the area to search and the search radius.

The Platform can control the AI vehicles in FSX (aircraft mainly used for simulating traffic) by giving it a set of waypoints which it must go through [5].

In order to avoid missing a waypoint, resulting in the aircraft going around to go through it, the path implementations must allow enough space for the aircraft to turn correctly.

C. Search Patterns

Although many search strategies exist, in this work only the Parallel Track, Spiral and Follow Shape patterns were considered. These patterns were implemented in a way to avoid non-covered area, even if at the expense of more time spent and overlapping areas. A more detailed explanation of the patterns is described in the following sections.

1) Parallel track

This pattern consists in following multiple parallel lines in alternate directions, as shown in Fig. 5. That being said, the only source of misspent efforts are the turns performed by the aircraft, which means that the turn radius of the aircraft greatly affects this search pattern.

To minimize the time spent during mission execution, the delimitation area is studied in order to discover the best angle to reduce the number of turns performed. This is made by discovering the bounding box with the shortest height. The turns shape also depends on the aircraft turn radius and search radius, having the following characteristics:

- If the turn radius is larger than the search radius, then the turn will look like a horseshoe.
- If it is equal or similar, the shape will be a semi-circumference.
- If it is smaller, the turn will be similar to an elongated semi-circumference.

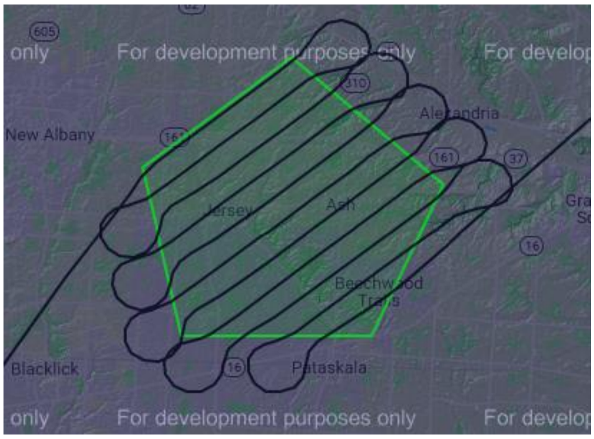


Fig. 5: Parallel Track being performed in a Pentagon shaped area, using a Cessna Skyhawk aircraft

2) Spiral In/Out

This pattern consists in performing a spiral, either inward or outward, until the area is totally covered, or the aircraft cannot turn more than it needs. As can be seen in Fig. 6; this strategy has the problem of covering area outside the area to search. The center of this pattern is the center of the area.

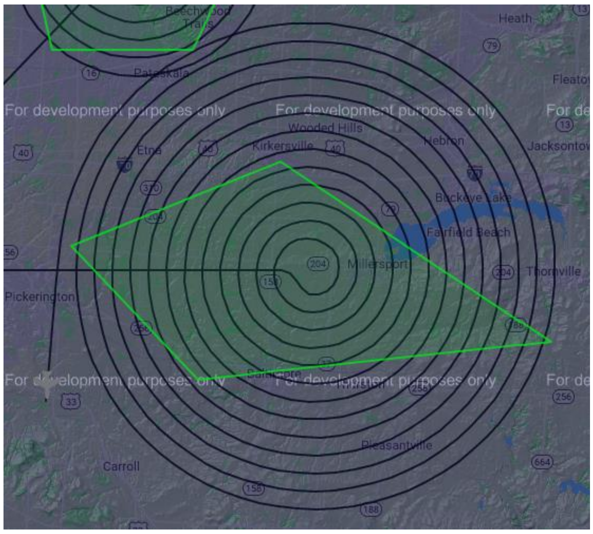


Fig. 6: Spiral pattern being performed in a Flatten area, using the vehicle Cessna Skyhawk.

3) *Follow Shape In/Out*

This pattern consists of replicating the area shape until it becomes a circle with a radius equal to the vehicle's minimum turning radius, as shown in Fig. 7.

To avoid non covered area, the number of spirals performed depends on the distance from the center to the furthest point in the mission area. This means that, when replicating the boundary furthest point, the distance between spirals will be equal to the search radius, while the distance between lines will be smaller than the search radius in the rest of the pattern, creating overlapped area (for non-circular area shapes). This effect is observable in Fig. 7.

Fig. 7: Follow Shape pattern being performed in a squared area, using the vehicle Cessna Skyhawk.

To perform the turns in the vertices of the boundary area, a slight adjustment is done to the pattern, which results in the erasure of sharp turns impossible to be performed by the aircraft. Although this adjustment exists, if the angle of the vertex is very small and the aircraft does not have enough space to turn, it will not be able to reach certain waypoints, turning back to do so, and thus decreasing its efficiency.

D. *Metrics*

In this project, the considered metrics were mainly the area covered, area overlapped, time spent, and fuel spent. Other metrics were studied but were not considered important due to the way the missions were implemented, such as:

- Average Speed: It is always the same as the aircraft's cruise speed. This happens due to imposing that velocity on the generated waypoints.
- Distance covered: It is calculated using the time spent and average speed. Since the speed is always the same throughout the mission, this metric ends up being similar to the time spent in terms of comparison. Time spent is more useful as a metric than distance covered so distance covered was ignored.
- Speed Efficiency: The average speed divided by the cruise speed, it is always 1, therefore not interesting to study.
- Distance Efficiency: Calculated by using the minimum theoretical distance to perform the search pattern. This minimum theoretical distance is the distance covered by the aircraft if it was not under-actuated. This metric is only interesting to study in the Parallel Track pattern, since on the others the aircraft follow the same path as if it were not under-actuated.

The time spent was calculated using the FSX's in-game time when the aircraft enters the mission area and when it ends the pattern. The area covered and overlapped were calculated through the use of the measuring points. The various measuring points were united to form lines, which were then transformed into trapezoids of height equal to 2 × *searchRadius* and center in the middle of the line, as represented in Fig. 8.

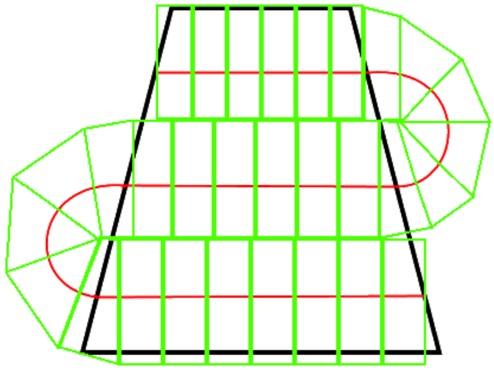


Fig. 8: Representation of the triangles and trapezoids made in order to calculate the area covered and overlapped.

The covered and overlapped areas are discovered using the GPC library [14], which allows Boolean algebra using polygons, such as union and intersection. The area covered was calculated using the area resultant from the union of all the rectangles and intersecting it with the mission area. The overlapped area was calculated by uniting the intersection of the rectangle with the covered area (before uniting the rectangle with said area) and intersecting it with the search area in the end.

IV. Experimental Setup

The results were obtained through various experiments combining five different area shapes, two vehicle types, two search radius and the three search patterns (five when considering the outward and inward variations). The areas have the shapes depicted in Fig. 9 in an attempt to change the shape to create a disparity between height and length (making it an elongated area) and creating different isoperimetric quotient values. Isoperimetric Quotient (IQ) [10] is a measurement given by the equation (1) that correlates the perimeter of an area with its size in a way that higher values indicate an area shape similar to a circle, while the opposite indicates irregular shapes. The values of this quotient for each area is given in Table I.

$$IQ = \frac{4\pi A}{p^2} \quad (1)$$

TABLE I: Areas Isoperimetric Quotient

Area	Isoperimetric Quotient
Squared Area	0.785
Regular Polygon	0.854
Flatten Area	0.413
Small Area	0.532
Circular Area	1

The used aircrafts were the Cessna Skyhawk and the Beechcraft Baron, being the former faster but having a larger turn radius.

The turn radius of the aircraft at cruise speed, as well as their cruise speed, are shown in Table II. The search radius was set to be either 1000 or 2000 meters.

Fig. 9: Search areas used in the experiments performed.

TABLE II: AIRCRAFT SPECS

Aircraft	Turn Radius	Cruise Speed
Cessna Skyhawk 172SP	984 m	56.59 m/s
Beechcraft Baron 58	2534 m	90.54 m/s

V. RESULTS

A. Distance Efficiency in a Parallel Track Pattern

The distance efficiency in a Parallel Track strategy gives an indication of the percentage of useful path, i.e. path not used to turn, compared to all the distance covered during the mission.

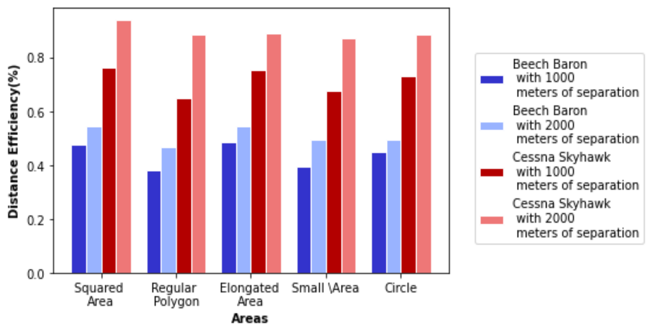


Fig. 10: Distance efficiency in a parallel track pattern.

As shown by Fig. 10, this pattern presents minimal variation between the different areas in terms of redundant movement. That happens because the number of turns needed is always given by the number of lines crossed minus one, which means that the height of the original bounding box has almost no impact on the metric. The used aircraft does however change the distance wasted due to the great difference in turn radius, making the Cessna Skyhawk more efficient than the Beechcraft Baron in that regard. The used separation also affects the metric by reducing the turn size compared to the lines. This change is more noticeable when using the Skyhawk since the turn actually changes shape.

B. Area Overlapped and Area Covered

The area overlapped in a Parallel Track search pattern is almost zero in all experiments while the Spiral pattern has an average of 3% and the Follow Shape differ a lot between experiments. This metric differs between experiments in the Follow Shape pattern due to two reasons:

- If the shape of the area creates a corner which the distance to the center is considerably bigger than the average distance of the points in the boundary to the center, this will result in a large amount of area overlapped in order to be able to cover all the area.
- Due to imperfections of the algorithm used, when turning in narrow angles, the agents miss waypoints, leading to complex maneuvers to get back on track.

The area covered in a Parallel Track is almost 100%, while the other patterns are quite close to this value. The reason for the other patterns to not present 100% area coverage is due to the circle they leave in the middle of the area, due to not being able to turn more than its minimum turn radius. This problem could however be corrected by either by lowering the speed in the end or by adding a certain path to cover the area.

C. Patterns Comparison

Fig. 11 seems to indicate that the best pattern for a squared shape area depends on the aircraft used, when considering the time spent as a metric. In fact, while the Parallel Track pattern seems to be the best one for the Cessna Skyhawk, the same does not happen for the Beechcraft Baron. This happens due to the difference in the turn radius between both aircraft, which, due to the nature of the Parallel Track pattern, leads to a massive waste in time due to turning.

It is also possible to discover two interesting results:

- The speed of the Baron seems to compensate the increased path distance in the same rate, making the time spent equal to when using the Cessna Skyhawk.
- All the other four patterns seem to have little difference in terms of time spent.

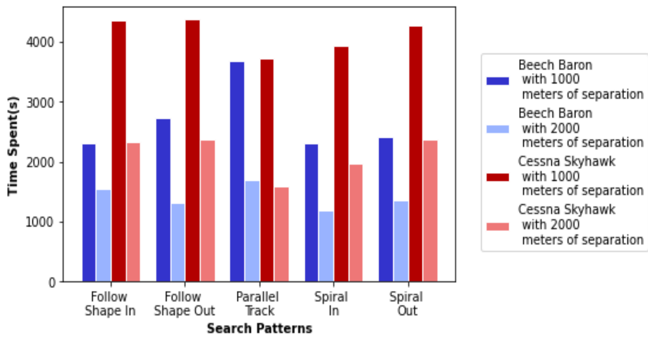


Fig. 11: Time spent in a Squared shaped area.

Fig. 12 confirms the idea that the Parallel Track pattern is the best one to handle areas with low IQ value. Although the Follow Shape pattern seems to be greatly affected by the irregularity of an area, this is result is due to a poor implementation of the algorithm when performing narrow turns, as it is explained in Section V-B.

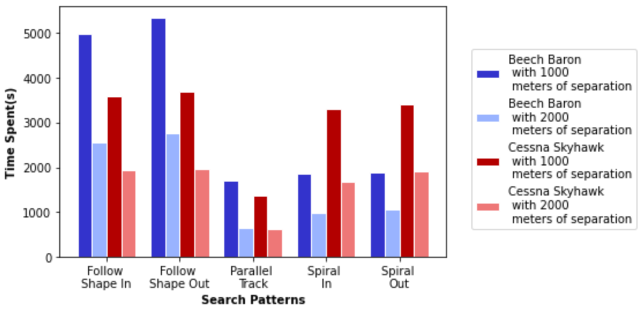


Fig. 12: Time spent in an Elongated area.

Fig. 13 confirms that when performing a measure mission in a circle-shaped area, the Spiral search pattern is the best no matter the parameters used.

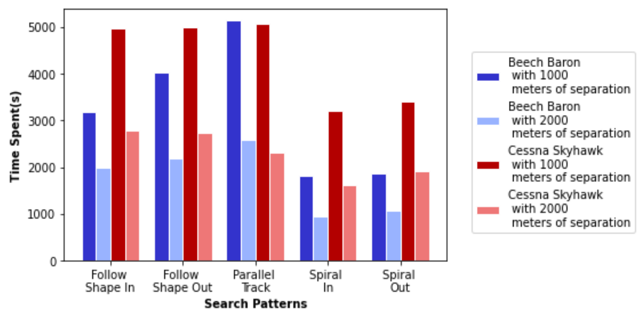


Fig. 13: Time spent in a Circle shaped area.

D. *Aircraft Comparison*

Albeit the Beechcraft Baron tends to be better in all search patterns while being equal to the Cessna Skyhawk in the Parallel Track, this result is only valid when considering the time spent metric. If we consider the fuel spent, then the result vastly tips in favor of the Cessna Skyhawk, as we can see in Fig. 14.

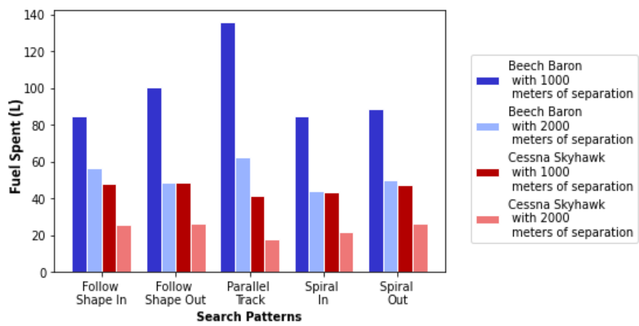


Fig. 14: Fuel spent in a Squared shape area.

E. *Results Conclusion*

By analyzing all the charts at the same time, it is possible to infer some interesting connections:

- Separation (search radius) is the parameter that affects time spent the most.
- Spiral In/Out seems to be better than Follow Shape In/out the bigger the isoperimetric quotient is.
- The lower the IQ value is, the better the Parallel Track search pattern is, when compared to the other patterns.
- The Beechcraft Baron is the best aircraft when considering the speed of a mission but is by far worse than the Cessna Skyhawk when considering fuel consumption.

VI. Conclusion and Future Work

An experimental study was conducted on area search patterns using aircraft in a simulation platform. This study allowed some conclusions to be drawn regarding the most effective search pattern to use depending on the area size and shape, as well as aircraft characteristics.

Despite the interesting and useful results, improvements can be made to the search patterns in order to be more open to the intent of the user. For instance, the Follow Shape search pattern tries to maximize area coverage, while spending more time in the process. The user might prefer less area covered in order to minimize the time spent. Besides that, other search patterns can be implemented for a richer study of the various strategies. More metrics should be implemented in the future to evaluate the missions depending on user preferences. This work can also be improved to allow the creation of prediction models, which could try to predict the best search pattern considering the scenario and user preferences.

Even though the experimental setup can be improved, as mentioned before, the current results are quite promising, and point to the possibility of improving efficiency of several types of tasks by using the rules inferred from this and similar studies, or by simulating the task at hand prior to its execution.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Tiago Carvalho was responsible for the software implementation, investigation and experiments, as well as results analysis and preparation of the paper. Daniel Castro Silva and Leonardo Ferreira were responsible for the concept, methodology design, supervision of the developments and editing of the paper.

References

[1] S. Waharte and N. Trigoni, "Supporting search and rescue operations with UAVs," in *Proc. International Conference on Emerging Security Technologies*, 2010, pp. 142-147.

[2] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 46-56, Sept. 2012.

[3] A. A. A. Alkhatib, "A review on forest fire detection techniques," *International Journal of Distributed Sensor Networks*, vol. 10, no. 3, pp. 1-12, 2014.

[4] T. Jackson and D. E. L. Vine, "Mapping surface soil moisture using an aircraft-based passive microwave instrument: algorithm and example," *Journal of Hydrology, Special Issue on Soil Moisture Theories and Observations*, vol. 184 no. 1, pp. 85-99, 1996.

[5] D. C. Silva, R. Silva, L. P. Reis, and E. Oliveira, "Agent-Based aircraft control strategies in a simulated environment," in *Agents for Games and Simulations – Trends in Techniques, Concepts and Design, Ser. Lecture Notes in Computer Science*, F. Dignum, J. M. Bradshaw, B. G. Silverman, and W. A. V. Doesburg, Eds., Springer Berlin/ Heidelberg, May 2009, vol. 5920, pp. 190-205.

[6] M. Bosse, N. Nourani-Vatani, and J. Roberts, "Coverage algorithms for an under-actuated car-like vehicle in an uncertain environment," in *Proc. IEEE International Conference on Robotics and Automation*, 2007, pp. 698-703.

[7] M. Benjamin, H. Schmidt, and P. Newman. (2019). MOOS-IvP documentation. [Online]. Available: https://oceanai.mit.edu/ivpman/pmwiki/pmwiki.php

[8] Y. Goh and S. Fan, "Path planning for AUV area coverage mission based on MOOS-IvP," in *Proc. IEEE Underwater Technology*, April 2019, pp. 1-10.
[9] A. C. Arbeit, "Adaptation and automation of search and rescue patterns with implementation for an operational unmanned aircraft system," Master's thesis, Department of Aeronautics and Astronautics, University of Washington, Washington, United States of America, 2013.
[10] H. Seyyedhasani, J. S. Dvorak, and E. Roemmele, "Routing algorithm selection for field coverage planning based on field shape and fleet size," *Computers and Electronics in Agriculture*, vol. 156, pp. 523-529, 2019.
[11] I. H. Osman, "Meta strategy simulated annealing and tabu search algorithms for the vehicle routing problem," *Annals of Operations Research*, vol. 41, no. 4, pp. 421-451, 1993.
[12] Microsoft Corporation. (2008). ESP SDK Overview. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/microsoft-esp/cc526948
[13] Microsoft Corporation. (2008). SimConnect SDK Reference. [Online]. Available: https://docs.microsoft.com/en-us/previous-versions/microsoft-esp/cc526983
[14] A. Murta. (2014). Gpc – general polygon clipper library. [Online]. Available: https://web.archive.org/web/20200613190343/http://www.cs.man.ac.uk/~toby/alan/software/

Tiago Carvalho was born in Braga, Portugal in 1997. He holds a Master's degree in Informatics Engineering from the University of Porto, concluded in 2020. He is currently a PhD candidate at the University of Porto, attending the Doctoral Program in Informatics Engineering.

Leonardo Ferreira was born in Aveiro, Portugal in 1995. He holds a Master's degree in Informatics Engineering from the University of Porto, concluded in 2018. He is currently a PhD student at the University of Porto, attending the Doctoral Program in Informatics Engineering. Currently, he is focused on artificial intelligence, recommender systems and cybersecurity but he already worked with robotics and simulation systems.

Daniel Castro Silva was born in Porto, Portugal in 1983. He earned a 5-year degree in Informatics Engineering in 2006 from the University of Porto and a PhD in Artificial Intelligence in 2011 also from the University of Porto. He develops his research at the Artificial Intelligence and Computer Science Laboratory (LIACC) at the University of Porto, and his main research field is artificial intelligence, in particular agent-based simulation.
Dr. Silva is an Assistant Professor at the Faculty of Engineering, University of Porto since 2013, having previously taught at the University of Coimbra between 2011 and 2013. He has participated in several European and national research projects, mainly focusing on artificial intelligence and intelligent simulation. Over the years, he has published over 18 journal articles and more than 50 conference papers. He has supervised over 35 MSc. thesis and is currently supervising 3 doctoral students.