# Multi-objective optimization at the EDge for Online and Real-time self-Adaptation of Autonomous vehicles

Flecheau Evan
*Lab-STICC*
*Université de Bretagne Occidentale*
Brest, France
evan.flecheau@outlook.fr

Lemarchand Laurent
*Lab-STICC*
*Université de Bretagne Occidentale*
Brest, France
laurent.lemarchand@univ-brest.fr

Dezan Catherine
*Lab-STICC*
*Université de Bretagne Occidentale*
Brest, France
dezan@univ-brest.fr

## I. Introduction

Autonomous vehicles are well-known for automatized tasks that are difficult or dangerous to be performed by humans. However, the environments in which those autonomous vehicles are evolving are generally dynamic environments that are hardly predictable. Thus, the challenge is for these to be following a predefined mission while adapting to their shifting environment in real time as efficiently as possible. Their mission often implies path planning problems, where self-adaptation of terrain modifications is required to finish a mission; and optimization of contradictory objectives, such as safety, risk assessment, travelling time or distance, consumed energy, etc. We choose to focus on supervision/monitoring missions (covering area with a lidar, with pictures, searching, etc) with two objectives : travelled distance (that could later be modeled into time or energy consumption) and covered area. We propose a multi-objective optimization (MOO) framework for a self adaptation of autonomous vehicles, with an offline/online approach, in order to solve covering/monitoring missions of autonomous vehicles. The offline process will predict a path that the autonomous vehicle will be initialized with, and the online process will be useful for the dynamic path replanning when obstacles will be detected. Our results show the interest to pre-compute routes offline. Such an approach has already been introduced and promises to be effective. For instance, Hu et al. [3] applied this strategy to another MOO and real-time path planning problem for Autonomous Surface Vehicles (AV). MOO approaches contrast with single objective ones as Niu et al. [6]. They focused on a path planning approach aiming at improving the endurance of an USV by optimizing its energy consumption. In [7], Zhang et al. also deal with USV path planning, proposing an hybrid genetic algorithm.

## II. Approach

In our model, the autonomous vehicle (AV) will be conducting the covering/monitoring missions in simulated maps of ports, represented as connected graphs. Obstacle detection is possible due to the simulated on-board lidar, and trajectory way-points describing the AV's path are generated with the Pareto Archived Evolution Strategy (PAES) algorithm [5], a popular Multi-Objective Evolutionary Algorithm (MOEA), one of the simplest algorithm capable of generating a Pareto set. Covered points are those in the range of the lidar during the tour. Trade-offs are to be found with the length of the tour, our second objective.

Our approach for solving a multi-objective optimization path planning problem is based on the hypothesis that, since MOEA are costly in resources, we may have the possibility to relieve the AV from some computational time by exploiting known environment data. Also, the PAES approach to MOO leads to a potentially re-usable archive of the best solutions calculated offline. Our idea for the embedded collision avoidance process is that we could re-use this archive to maintain a fully optimized intermediary path until the mission is completed. We would then have a bi-phase offline/online approach to this problem, where a first path would be generated offline, while the online part would be used to correct the path when dynamic obstacles were to intersect with the AV's initial tour.

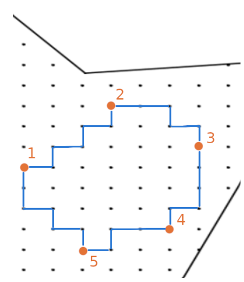


Fig. 1. Illustration of a path planning mission for the AV. All 5 way-points have to be followed in a specific order. Obstacle events could occur and intermediary routes would be calculated to avoid collisions, while visiting initial way-points if possible.

The offline process is split into three sub-processes :

1) Data and graph initialization for all shortest paths using Floyd-Warshall's algorithm.
2) Non-dominated solutions generation using PAES algorithm.

3) Best solution selection using TOPSIS algorithm.

The two first steps are realized with the same approach as [1], using PAES. TOPSIS [4] is a famous Multi-Criteria Decision Making algorithm that has been used in many fields [2]. Its execution time (shown in Table 1) and algorithmic complexity justify our choice for the decision making and we use it in our offline process to select the best optimized route for the AV.

The online process uses Dijkstra's algorithm and is executed every time an obstacle must be avoided (modeled as invalidated points in the graph). It recomputes shortest paths in the updated graph from current node to next way-point. This method ensures the validation of the mission, but doesn't take directly into account the trade-off to be found between distance and covering for the tour.

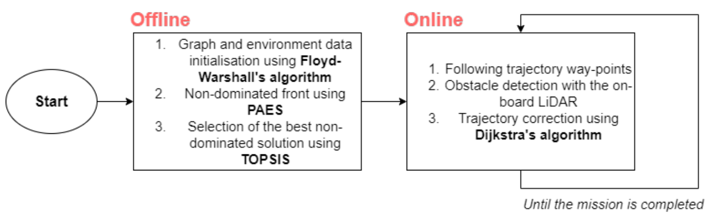


Fig. 2. 2-phases path planning strategy

## III. EXPERIMENTS

This section is about scenario experiments that we have conducted to justify the relevance of the previous system (Fig. 1). A simulation of the first path planning version for a full mission is composed as following :

1) grid parameters and the lidar range of the AV are used to generate a connected graph on the map and visibility between points.
2) PAES produces a set of routes for a given starting node (Pareto non-dominated solutions).
3) The TOPSIS algorithm is used to choose a route and initialize the AV mission.
4) Random obstacles are then generated, and Dijkstra's algorithm corrects compromised way-points.

We performed multiple simulations with different grid grains (shown in Table 1). The main purpose of these tests are to show the interest of an offline/online approach to path planning in terms of trade-offs exploration (not shown here, but assessed in [1] for the offline phase) and in terms of computational effort saved for the online process. It is also a first attempt at verifying the impact of a trajectory update on distance and covering metrics.

As shown in Table 1, execution times for PAES and Dijkstra's algorithms depend on the graph's parameters : the more complex the graph is, the more time is required for these two algorithms to finish processing. However, TOPSIS has fairly constant execution times, whatever the map it is tested on. It is due to the fact that this algorithm is always applied to the archive, i.e. a fixed amount of solutions. There are 3 orders of magnitude between the offline (PAES) and online (Dijkstra) computations, justifying the offline phase.

TABLE I
MEANS OF EXECUTION TIMES OF 20 TESTS CONDUCTED ON THE BREST PORT MAP, WITH A 100 TO 25 METERS DISTANCE BETWEEN POINTS AND A 150-METER RANGE LIDAR. PROCESSOR USED FOR EXECUTION : INTEL(R) CORE(TM) I7-8700K CPU @ 3.70GHZ, 4 GB RAM

| Map : BREST port | | | | |
|---|---|---|---|---|
| Rows | Columns | PAES (sec) | TOPSIS (ms) | Dijkstra (ms) |
| 50 | 50 | 7.32 | 0,030 | 1.4 |
| 55 | 55 | 9.91 | 0,028 | 2.6 |
| 60 | 60 | 14.17 | 0,027 | 2.3 |
| 65 | 65 | 16.75 | 0,033 | 4.7 |
| 70 | 70 | 23.06 | 0,031 | 5.1 |
| 75 | 75 | 26.71 | 0,031 | 10.3 |
| 80 | 80 | 33.44 | 0,029 | 10.4 |

However, resulting tours (not presented here) present deceptive characteristics: despite having decent results on small obstacles, large obstacles often force the AV to take non optimized routes in order to complete its mission. This is explained by route updates through already explored zones, resulting in both a worse distance for the tour and worse covering rates. A second version of the algorithm is then considered : we would adapt our PAES algorithm in an online scenario, updating online the alternative tours in the embedded archive instead of running PAES again from scratch, allowing to efficiently and rapidly reroute when obstacles arise.

## IV. CONCLUSION

This study deals with an MOO and dynamic path replanning problem, in the context of an AV assigned with a covering/monitoring mission while avoiding collisions with unpredictable obstacles. We adopted a bi-processes offline/online approach, with computational effort on the offline phase. This allows to decrease the response time when avoiding obstacles online. As future works, we plan to target an embedded platform for the selected online phase implementation.

## REFERENCES

[1] ET AL., H. O. Energy-based usv maritime monitoring using multi-objective evolutionary algorithms. *Ocean Engineering 253* (2022), 111182.

[2] ET AL., M. B. A state-of-the-art survey of topsis applications. *Expert Syst. Appl. 39* (2012).

[3] HU, L., AND ET AL. Colregs-compliant path planning for autonomous surface vehicles: A multiobjective optimization approach**the authors should like to thank innovate uk, grant reference, tsb 102308, for the funding of this project. *IFAC-PapersOnLine 50*, 1 (2017), 13662–13667. 20th IFAC World Congress.

[4] HWANG, C.-L., AND YOON, K. *Methods for Multiple Attribute Decision Making*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1981, pp. 58–191.

[5] KNOWLES, J. D., AND CORNE, D. W. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation 8*, 2 (2000), 149–172.

[6] NIU, H., LU, Y., SAVVARIS, A., AND TSOURDOS, A. An energy-efficient path planning algorithm for unmanned surface vehicles. *Ocean Engineering 161* (2018), 308–321.

[7] ZHANG, W., XU, Y., AND XIE, J. Path planning of usv based on improved hybrid genetic algorithm. *2019 European Navigation Conference (ENC)* (2019), 1–7.