Multi-Objective Optimization for an Online Re-Planning of Autonomous Vehicles

Kilian Le Gall, Laurent Lemarchand, Catherine Dezan

Multi-Objective Optimization for an Online Re-Planning of Autonomous Vehicles

Kilian Le Gall
Lab-STICC/UBO
Brest, France
kilian.legall1@etudiant.univ-brest.fr

Laurent Lemarchand
Lab-STICC/UBO
Brest, France
laurent.lemarchand@univ-brest.fr

Catherine Dezan
Lab-STICC/UBO
Brest, France
catherine.dezan@univ-brest.fr

***Abstract*—Autonomous vehicles are well-known for automated tasks that are difficult or dangerous to be performed by human. However, the environment in which those Autonomous Vehicle (AV) are evolving are generally difficult to predict. Thus for these, the challenge is to achieve a predefined mission while adapting themselves to their shifting environment in real time as efficiently as possible. Their mission often includes path planning problems, where self-adaptation to terrain modifications is required while maintaining contradictory objectives, such as safety, risk assessment, travelling time or distance, consumed energy. We choose to focus on supervision 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. The offline process will predict a path that the autonomous vehicle is initialized with and the online process will be useful for the dynamic path re-planning when obstacles are detected. Our results demonstrate the benefits of reusing the offline pre-computed solutions for the online phase and for dynamic path re-planning.**

***Index Terms*—drone/AV path planning, embedded decision making, obstacle avoidance**

I. Introduction

Autonomous vehicles obtained a lot of attention these recent years. Indeed, they are able to perform dangerous or difficult tasks, such as monitoring hardly accessible areas, efficiently collecting data or conducting stealth oriented tasks without human supervision. Path planning optimization is a really important topic since it allows autonomous robots or vehicles to carry on their mission even when unexpected events happen. Moreover, when dealing with autonomous vehicles, it is near mandatory to consider other optimization issues related to path planning optimization. Energy consumption for instance, is an important aspect of autonomous vehicles, as much as safety, risk assessment, travelling time, collision avoidance, etc. Therefore, dealing both with path planning optimization and multiple objectives appears to be a real challenge.

This work is granted by joint Thales/Lab-STICC LATERAL lab.

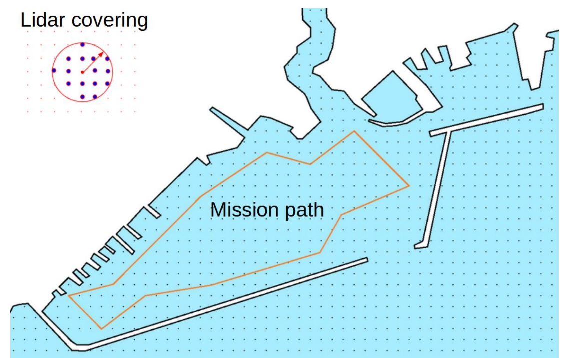

Figure 1. Path for monitoring mission with covering by a lidar

We plan to take benefit of the routes computed offline for obstacle avoidance. Hu et al. [5] use such an approach for another MOO and real-time path planning problem for Autonomous Surface Vehicles (AV). MOO approaches contrast with single objective ones such as Niu et al. [7]. They focus on a path planning approach aiming at improving the endurance of an Unmanned Surface Vehicle (USV) by optimizing its energy consumption. In [9], Zhang et al. also deal with USV path planning, proposing an hybrid genetic algorithm. Previously, we also formulated the problem of monitoring with shortest path as a Mixed Integer Programming Problem in [8] and solved it with meta-heuristics in [2], but these works do not include an online phase for dynamic environment management.

Regarding the multi-objective optimization (MOO), we focus on the travelled distance (*Length* correlated with energy consumption and time) and the covered area by the lidar of the AV (*Cov* correlated with the surface not covered), as illustrated in Fig. 1. We have chosen these two objectives. The consumed energy is an important factor to achieve the whole mission of autonomous vehicles. Their capacity to conduct a mission from start to end while applying path modifications along the route is strongly affected by their energy consumption, specially when dealing with unexpected re-planning of the route. The covered area is the surface of the supervised area defined by the autonomous vehicle with its on-board detection device. Regarding path planning

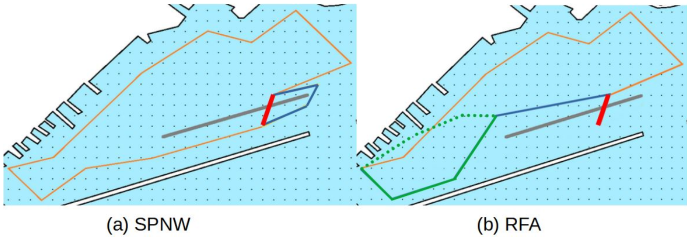

Figure 2. Obstacle avoidance : (a) repairing with SPNW (blue line) (b) swap to another solution (green) with RFA (closest point connection in blue)

optimization, since AVs are supposed to be self-adaptive, they need to adapt themselves if they encounter any unexpected obstacle, e.g. another ship.

During the offline phase, an evolutionary algorithm (e.g. PAES [6]) generates an archive of feasible solutions to initialize the mission. The solutions are represented on a graph and defined by a series of waypoints that trace the path to be taken. Each solution in the archive is *non-dominated*, i.e. no other solution in the archive performs better for both objective functions (*Length* and *Cov*). Then a multi-criteria decision analysis method such as TOPSIS [3] selects one solution in the archive to launch the mission. Once the mission is launched, the other solutions in the archive are no longer used.

During the online phase, the path taken by the drone could involve a risk of collision with obstacles that were not taken into account by the solution generator. One solution that has already been implemented [4] includes the computation of the shortest path while taking into account the obstacle (Shortest Path to Next Waypoint (SPNW) which uses Dijkstra algorithm [1]). However, this solution only takes into account the route between the two waypoints affected by the obstacle, and not the drone's overall mission. As a result, the obtained trajectory may cause the drone to revisit areas that it has already visited or will soon visit, resulting in useless travel.

II. Approach

In this study, we propose two approaches for obstacle avoidance in drone missions.

The first approach is the implementation of SPNW, while the second approach involves Recovering the mission From the Archive (*RFA*) reusing the archive generated by the offline phase to generate a new archive. This new archive is composed from two parts. The first one from the already travelled path, and a second one from the archive's solution. Both are illustrated in Fig. 2. We can see that the strategy SPNW in 2(a) forces the AV to go backward to bypass the obstacle, extending the travelled path. With the RFA algorithm in 2(b), a new solution is proposed saving distance while keeping a good covering area. We evaluate the performance of these two approaches in terms of computation time and quality of the obtained solution.

In a previous work [4], the shortest path algorithm (SPNW) was implemented and applied on a graph representing the area to be monitored degraded by obstacles. This enables the drone to find new routes when it detects a potential collision. This same version has been adapted to consider both objectives: Length and Cov are two factors that should be minimized. Length should be as short as possible, while Cov refers to the surface area that is *not* covered and should also be minimized.

The approach presented in this paper aims to reuse solutions present in the archive but not selected for the mission launch. By combining segments of previously calculated solutions with the already traversed part of the launched solution, new solutions are obtained. These solutions propose paths to avoid the obstacle which are alternatives to the SPNW solution.

It could then be possible to bypass the obstacle while avoiding traversing backward in ever monitored areas and thus keeping the mission as optimized as possible. This strategy requires less computation than running the offline process again with an embedded software. Incorporating this would also eliminate the necessity of integrating the generation of solutions offline.

This strategy can also be optimised, by limiting the workload. We propose to embed only a fragment of the archive to limit the computation time.

A. Offline Solution generation

In our model, the autonomous vehicle carries out monitoring missions in area illustrated by port maps, which are represented as a connected graph over a grid where cardinal neighbours are connected. The drone's trajectory is modeled as a chromosome (like in [2]), consisting of the initial node and the various nodes (waypoints) that the drone will traverse. The objective functions to be minimized are calculated by following the chromosome's path and using the graph representation of the area (see Fig. 1).

The PAES algorithm manipulates these chromosomes with mutations. A mutation occurs every generation and the number of generations is set at the start of the algorithm. The archive that stores the non-dominated solutions is of a fixed size. PAES policy flavours diversity of solutions in terms of objective values in the archive [6]. The different mutations implemented in our model are the following : gene addition, gene replacement, gene deletion and gene swap. Each mutation has a probability to occur and only one mutation is allowed during a generation.

At the end of this offline phase, the path that will be used for the mission is then selected among the solutions in the archive by a multi-objective decision algorithm (TOPSIS, [3]).

B. Online phase RFA

Once the mission is launched, obstacles can arise, such as a ship passage. The obstacle is the representation of the semgent of the map that will not be available during the mission. The corresponding set of nodes in the graph of the area is considered as the obstacle for the drone.

By comparing the nodes traveled between the different waypoints of the path with the nodes composing the obstacle, it is possible to determine if the mission's trajectory hits an obstacle. In this case, the RFA algorithm is triggered.

The RFA method naturally requires more computation since it includes, for each produced solution, a second correction thought SPNW of the trajectory in case this new solution also crosses the obstacle. In addition to the production of solutions and their selection by TOPSIS, the computation time varies depending on the size of the archive produced by PAES.

1) Repairing: As illustrated in Fig. 2 (b), the repairing proceeds as follows. Let p be the index of the waypoint w_p in the current solution C next to the obstacle. For each solution S of the archive A, the algorithm searches for waypoint w_q in S the closest to w_p. It then builds a solution $S' = (w_1, ..., w_p, w_q, ..., w_n)$, with waypoints 1 to p from C, and waypoints q to n from S.
S' is evaluated added to the repaired archive A', while checking for non dominance. It is discarded if dominated by a member of A'.

Similarly to the offline choice of the mission launched in A, the selection of the new current solution is performed using TOPSIS over A'.

2) Workload policy: Embedding a partial subset of solutions allows to limit online computational effort. This workload is expressed as a percentage of the archive embedded and used by RFA. The selection policy is as follows. The whole archive is sorted according to one of the two objectives (Length or Cov) and solutions are picked up at a frequency corresponding to the desired workload. This policy aims to preserve the diversity of the selected subset. Example: for 10% of an archive of 100 solutions, embedding will be composed of one solution every 10. This strategy is named RFA100:10 according to the archive size and the workload percentage.

III. Experiments

We set up an experiment to measure the effectiveness and compare our obstacle avoidance approaches. When an obstacle arises, we apply both SPNW and RFA, and we check whether the decision process maintains its current mission with obstacle avoidance from SPWN or switches to a repaired solution from RFA new produced archive. The strategies are compared according to the objective functions values.

The experiments are realized with those parameters :

- PAES Genetic algorithm generations : 40 000
- Archive size : 100 solutions
- Workload percentage : 100%, 90%, 80%, 70%, 60%, 50%, 40%, 30%, 20%, 10%

We have run 1000 test cases over the area depicted in Fig. 1. Randomized offline phase produces an archive A, and a mission is selected using TOPSIS. An obstacle crossing the corresponding path is then randomly generated (made of 2 connected points).

Both strategies SPNW and RFA are then applied in order to solve the obstacle avoidance problem. SPNW is one of the solutions of RFA new archive A'. We consider that RFA is useless for the test case if the solution chosen is the one repaired by SPNW (which is included in the RFA archive when a new solution is selected by TOPSIS). Conversely, if it is a new solution from the repaired archive (A'), we consider that RFA method is useful.

In order to explain TOPSIS choice, produced new solutions are compared according to Length and Cov objectives. Here, we want to know what percentage of RFA solutions are dominant compared to SPNW solutions. In cases where the chosen solution is from the repaired archive and is not dominant, we need to check what are the difference between the selected solution and the simple SPNW solution to determine why TOPSIS has chosen this new solution. The differences between the objective functions are expressed as a percentage evolution from the initial solution repaired with SPNW and the selected solution.

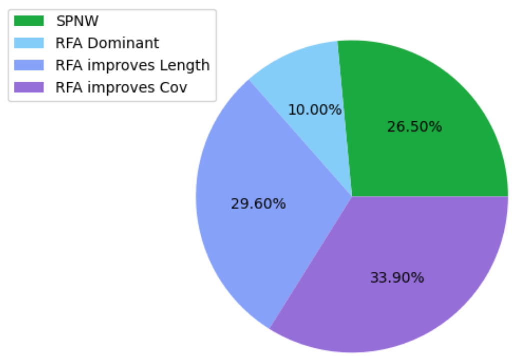

Figure 3. Solution choice between SPNW and RFA for 1000 test cases

The results in Fig. 3 show that the RFA archive repair method is selected through TOPSIS over the simple SPNW bypass method in 73.5% (RFA dominant 10.00% + RFA improves Length 29.60% + RFA improves Cov 33.90%) of the test cases. The 10.00% RFA dominant over the SPNW solution, are solutions with a shorter distance traveled for an even smaller uncovered area. In the remaining 63.50% (RFA improves Length and RFA improves Cov) of cases, TOPSIS chooses a trade-off solution from the archive, which means that either a decrease in distance was achieved at the expense of a larger uncovered area, or a greater distance was traveled for a smaller uncovered zone. When there is no dominant solution, TOPSIS selects a solution based on its ratio between the two objective functions, therefore there are solutions that prioritize minimizing the distance traveled and others that prioritize minimizing the uncovered surface. In situations where TOPSIS chooses a solution from the RFA archive that is not dominant (for a total of 29.60% + 33.90%, all non dominant solution from RFA strategy, as shown in Fig. 3), the explanation can be found in the value comparison between SPNW solution and RFA selected solution.

When RFA improves Length (which occurs in 29.60% of cases shown in 3), there is a reduction of 15.31% in Length and an increase of 5.01% in Cov (i.e decrease of the monitored area). Likewise, when RFA improves Cov (which occurs in 33.90% of cases in 3), there is a decrease of 3.76% in Cov ((i.e increase of the monitored area) and an increase of 11.13% in Length.

As stated in Section II-B, a selection policy could be required to manage the execution time. Thus, a study of the impact of the workload on both TOPSIS choice and execution time of RFA was carried out as shown in Fig. 4. The workload corresponds to the percentage of the archive used during RFA.

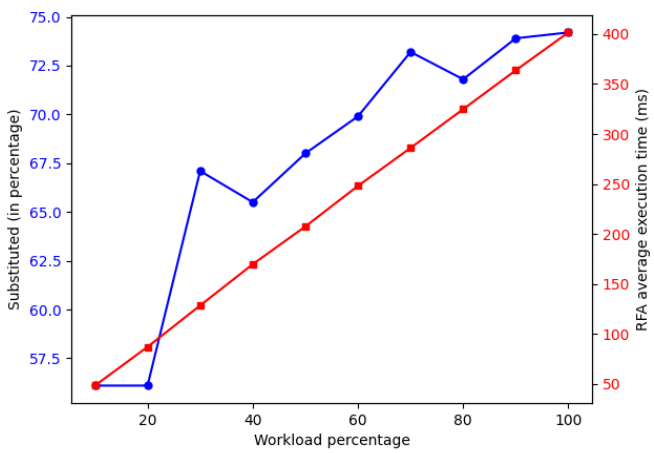

Figure 4. For 1000 test cases, (left axis, blue curve) percentage of solutions substituted to SPNW by RFA according to the workload (fraction of archive of 100 solutions provided to RFA) ; (right axis, red curve) computation time for RFA

Fig. 4 shows the use of RFA approach as compared to SPNW and associated computational effort. The Table I details the reasons of the choice of RFA solution for some workload values.

Table I
COMPARISON OF SPNW RESULTS AND EXECUTION TIME WITH RFA STRATEGIES FOR DIFFERENT RFA WORKLOADS FOR 1000 TEST CASES

	SPNW	RFA 100:100	RFA 100:50	RFA 100:10
Dominates SPNW	-	10.00%	9.00%	4.30%
Improves Length	-	29.60%	26.40%	6.10%
Improves Cov	-	33.90%	31.70%	43.40%
Total substituted to SPNW	-	73.50%	68.00%	56.10%
Execution time (ms)	2.12	401.55	207.57	48.51

On average for a 100% workload strategy, the method has a 73.5% solution substitution from SPNW and a computation time of 401.55 ms (see Table I RFA100:100). On average, a 10% workload strategy has a substitution of 56.10% and a completion time of 48.51 ms (see Fig. I RFA100:10). A division of the workload by 10 also divides calculation time by 10 but only decrease solution substitution from SPNW by 18.1%. However it also decrease the percentage of dominant solution to SPNW from 10.00% to 4.30% (see row dominates SPNW in Table I).

IV. Conclusion

In this paper, we proposed a new approach, RFA, for avoiding an obstacle by reusing an archive of existing solutions. We implemented this method and compared it to a simpler approach of finding the shortest path to bypass the obstacle. The obtained results gave a 73.5% of the tests in favor of RFA for quality results. However, it's calculation time is way superior to SPWN. A solution in the form of a partial workload is possible to decrease execution time to the detriment of the solution quality. From 73.5% with a 100% workload to 56.10% efficiency with a 10% workload. This type of trade-off offers good perspectives for an online re-planning during the mission.

Our final goal is to embed the software onto a drone, with real-time constraints on path computation times. It is necessary to conduct a study to determine the amount of time and memory required for this method, given that it is supposed to avoid collisions in real-time. In case the computation time is too long, we could explore the possibility of using only a subset of the existing solutions to reduce the time required for computation. To do this, we would need to analyze the selected solutions used for repair from the initial archive before transformation, in order to establish a model that would enable us to take only a portion of the archive and significantly reduce the time required for circumventing the obstacle with this method.

References

[1] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[2] Hand Ouelmokhtar et al. Energy-based usv maritime monitoring using multi-objective evolutionary algorithms. *Ocean Engineering*, 253:111182, 2022.

[3] Majid Behzadian et al. A state-of the-art survey of topsis applications. *Expert Syst. Appl.*, 39, 2012.

[4] Evan Flecheau, Laurent Lemarchand, and Catherine Dezan. Multi-objective optimization at the EDge for Online and Real-time self-Adaptation of Autonomous vehicles. Colloque du GDR SOC2, June 2022. Poster.

[5] Liang Hu and et al. Colregs-compliant path planning for autonomous surface vehicles. *IFAC-PapersOnLine*, 50(1):13662–13667, 2017. 20th IFAC World Congress.

[6] Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[7] Hanlin Niu, Yu Lu, Al Savvaris, and Antonios Tsourdos. An energy-efficient path planning algorithm for unmanned surface vehicles. *Ocean Engineering*, 161:308–321, 2018.

[8] Hand Ouelmokhtar, Yahia Benmoussa, Jean-Philippe Diguet, Djamel Benazzouz, and Laurent Lemarchand. Near-Optimal Covering Solution for USV Coastal Monitoring using PAES. *Journal of Intelligent and Robotic Systems*, 106(1):24, September 2022.

[9] Weicheng Zhang, Yanmin Xu, and Jinpeng Xie. Path planning of usv based on improved hybrid genetic algorithm. *2019 European Navigation Conference (ENC)*, pages 1–7, 2019.