Salima Bella*, Assia Belbachir, and Ghalem Belalem

A hybrid air-sea cooperative approach combined with a swarm trajectory planning method

https://doi.org/10.1515/pjbr-2020-0006
Received October 7, 2019; accepted December 17, 2019

Abstract: This work addresses the issue of ocean monitoring and clean-up of polluted zones, as well as the notion of trajectory planning and fault tolerance for semi-autonomous unmanned vehicles. A hybrid approach for unmanned aerial vehicles (UAVs) is introduced to monitor the ocean region and cooperate with swarm of unmanned surface vehicles (USVs) to clean dirty zones. The paper proposes two solutions that apply to trajectory planning from the base of life to the dirty zone for swarm USVs. The first solution is performed by a modified Genetic Algorithm (GA), and the second uses a modified Ant Algorithm (AA). The proposed solutions were both implemented in the simulation with different scenarios for the dirty zone. This approach detects and reduces the pollution level in ocean zones while taking into account the problem of fault tolerance related to unmanned cleaning vehicles.

Keywords: UAV, USV, swarm, trajectory planning, fault tolerance

1 Introduction

Swarming is a collective behavior observed in animals of similar size, such as insects that gather and move in groups. In the context of robotics, swarm robotics is an approach for coordinating multi-robot systems [1, 2]. Most of the research carried out in cooperation with dynamic unmanned vehicles such as air, surface, ground unmanned, etc., has focused on formation control, which refers to the task of controlling these unmanned vehicles to follow a predefined trajectory while maintaining the desired formation pattern [3].

Path planning refers to the search for an optimal or sub-optimal path from the starting position to the goal in the environment. Trajectory planning is based on two key technologies: environmental modeling and planning algorithms. A reasonable model of the environment can help to reduce the amount of search and space occupancy over time [4], the grid representation, the geometry of space law, the topological method and electronic chart are reasonable models [5]. Grid method characteristics are simple, easy to implement and can be applied to different algorithms. Trajectory planning algorithms commonly include two methods: traditional planning and intelligent planning methods. Traditional methods include: artificial potential field method [6], A* method [7], D* method [8] and so on. Intelligent planning methods include: genetic algorithm [9], colony algorithm [5], neural network [10], particle swarm algorithm [2], etc. In addition, there is a mixed planning method based on the combination between traditional and intelligent planning.

According to [9], heuristic algorithms that implement searches in the solution space can be classified as instance-based or model-based algorithms. The instance-based algorithms generate new candidate solutions using the current solution or a population of solutions, such as genetic algorithms. GA are iterative stochastic optimization algorithms. They provide solutions to problems that do not have computational solutions in reasonable time analytic or algorithmic time. Model-based search algorithms generate candidate solutions using a parameterized probabilistic model, updated according to previous solutions. It allows the search to focus on regions containing high quality solutions such as Ant Colony Optimization (ACO). ACO is a meta-heuristic modeling based on the ant’s drilling behavior. Ant System (AS) was the first ACO algorithm applied to the travelling salesman problem (TSP) [11].

This work aims to present a hybrid hierarchical approach for a better management and easier control between the different unmanned vehicles. Moreover, this
architecture allows fault tolerance and scalability compared to the self-organized approach that does not allow or hardly allows scalability with increased complexity for efficient management of its entities. Centralized management has a central node with deterministic decision-making capability and easy to implement coordination. This central node has a global view of the unmanned monitoring and cleaning swarm vehicles activities. Distributed management begins when the monitoring vehicle is assigned to a region. It has the ability to coordinate its cleaning swarms. These swarms plan their movement based on the proposed solutions to get to the dirty zones from their storage bases. These solutions propose a calculation method that quickly provides a workable solution to the planning problem. As this cleaning vehicle is a mechanical, electronic and computer system which may at some point fail at the hardware/software level, a solution is proposed to allow the replacement of failed vehicles by the competent cleaning vehicle during the realization of the cleaning mission.

This paper presents a hierarchical hybrid cooperative approach for heterogeneous unmanned vehicles. The approach includes two solutions to the trajectory planning problem for the unmanned surface vehicle (USV) swarm. The first one is based on the modified Genetic Algorithm, whereas the second relies on the modified Ant Algorithm. The proposed approach allows the cooperation of an unmanned aerial vehicle (UAV) to monitor an ocean region with an unmanned surface vehicles (USV) swarm to clean dirty oceanic zones. The UAV is assumed to be equipped with on-board sensors that allow it to locate dirty zones. The UAV discretizes its environment map and updates it with the collected information related to dirty zones. The UAV sends its environment map to its general coordinator (represented by a laptop and guided by a human operator). After an analysis of this collected data, the general coordinator allocates the explored map (way-points) to the USV swarm to clean each dirty zones. This swarm navigates to the assigned dirty zone and cleans it based on the proposed solutions. In addition, the proposed approach is complemented by a failure managing method of a USV during the execution of its cleaning task.

The remainder of this paper is organised as follows: Section 2 presents several related works; Section 3 describes the proposed approach for the different unmanned vehicles with the proposed solutions, and models them by logical formalization; in Section 4 the proposed approach is compared to the related works; Section 5 illustrates an example to simulate the operation of the proposed approach; finally, Section 6 concludes this paper and provides some future research directions.

2 Related works

Many heuristic and meta-heuristic algorithms have been applied to the path planning problem of unmanned vehicles, such as GA, ACO or PSO (more approaches can be found in [6]). In addition, several methods allow for coordination and control of multiple unmanned vehicles in a swarm or formation in trajectory planning [12] and to execute various applications such as cleaning, surveillance, rescue, detection and localization. For example, the CADDY cooperative orientation system which was proposed to monitor the behavior of human divers during the execution of missions at sea [13]. CADDY functionalities rely on the coordination of two robotic platforms, an Autonomous Surface Vehicle (ASV) and Autonomous Underwater Vehicles (AUV), and a guide. The cooperative path-following method is based on the virtual target approach. This method is developed in two stages (defined in [14, 15]). First, it relies on the consolidated single-vehicle virtual-target based path-following technique (using a Lyapunov function). Then, it extends the concept to a multi-vehicle system based on speed regulation, depending on the relative distance between the virtual targets (further details in [16]).

Thus, in [3], the authors introduced an improved swarm-based path-following guidance system for an autonomous multi-vehicle marine system. In the seminal idea, a team of USVs is required to join a formation using a potential-based swarm aggregation methodology, while a virtual target based guidance module drives the whole formation towards a desired reference path (further details in [16]). The proposed document improves the performance of the work [15] by presenting the modelling of the two following problems: the positive overvoltage speed (typical constraints of the maritime vehicles) and the saturation speed at a maximum value. Another work [17] proposed an organic computing approach to develop a complete framework for the design (observers) and control of autonomous collaborative robot swarms, with particular emphasis on quadcopters collaborating with each other to perform spatial tasks. The proposed approach has facilitated the adaptability and self-optimization of drone individuals, the optimization of collaborative efforts between drones and an efficient control of the swarm by the human user on several levels of abstraction [18].

Furthermore, a prototype [19] of a USV based on two gas sensors was developed to locate the gas source of an oil spill in the sea. The gas concentration received by the two sensors, applied to a fuzzy logic controller, determines the rate at which each propeller motor approaches the gas
target. The result shows that the USV prototype is able to approach and locate the target with average accuracy obtained when searching for the location of the gas source of 90.1%. Thus, a new robotic system swarm [20] was proposed to locate and pick up an oil spill on the surface of the water (ocean, river, lake.). A coordinator determines the position and the center of the spill using a GPS receiver, then a barge carrying the robots moves to the workplace. Depending on the state of the spill, the coordinator can send a swarm of robots to surround and collect the oil spill, then place the barge with oil suction equipment and move it to another location to remove the oil more safely. This new system saves money and time.

In addition, a USV design system [21] was proposed to detect and collect samples of oil spills. The manufactured USV uses a straight-line tracking algorithm with GPS navigation to localize the polluted zone. When it reaches the polluted zone, it activates a sampling mechanism to extract a sample from the water surface and executes the image-based detection algorithm using an automatic learning algorithm to confirm the oil presence. Experiments with this vessel have shown excellent maneuverability, a reliable mechanism for collecting samples and a reasonable accuracy in detecting oil contamination.

On the other hand, an optimal path-planning method for a USV was proposed in [5]. The authors used the grid method modelling for typical obstacles. This grid divides the workspace of the robot (environment) into cells. The white and black cells represent the free space and the obstacle space, respectively. The shortest path is obtained by this frame with the USV eight movements in the grid. The global path planning method used is the ant colony algorithm based on ACO. The roulette method is applied to select the next point at which the USV can reach.

Another ant algorithm ‘Max-Min Ant System (MMAS)’ was studied in [11] to find an optimal or near-optimal path, in which robots should explore the environment at the same time they plan the path. Thus, the authors analyzed the distance traveled by robots during the first iteration of the algorithm to analyze the quality of the solutions obtained. The environment is represented by a topological map. The paths are represented by a sequence of actions that robots should execute to reach the goal. MMAS provided a very good performance and obtained better results (best distance traveled) compared to the best distance traveled obtained by GA and best distance traveled obtained by A* algorithm.

Contrariwise, a real-time in-flight trajectory planner (RTTP) [22] of a fixed-wing UAV was presented for the volcanic monitoring and ash sampling task. The RTTP is based on a GA that allows that finds a a collision-free trajectory with less energy and is adapted to long and high altitude atmospheric conditions over the Volcán de Fuego in Guatemala. In [9], another method for planning pathway is applied, where GA is used to find a sequence of actions for autonomous mobile robots to perform to reach the goal. The robot does not know in advance the disposition of the environment and only has a rough estimate of the starting positions and objective. At first, a set of actions are generated randomly; robots execute these actions. After, their fitness is evaluated by the distance traveled and the Euclidean distance from the goal. Individuals are selected by tournament to breed. Then, a new sequence of actions is generated by applying crossover and mutation operators. The evolution continues only for the sequence of related actions that did not reach the goal. GA has a better average performance and the shortest distances traveled by the solutions returned by GA than the solutions returned by A* and its improvement (C*). In [12], another GA was proposed for the motion planning of heterogeneous holonomic robot swarms. This algorithm consists of a global path planner (GPP) and a motion planner (MP). The GPP algorithm searches for a path that the robot swarm center must move along in the free space of a Voronoi diagram (2D workspace). The MP is a GA based on an artificial potential field. The repulsion keeps robots away from obstacles and the ‘spring’ function maintains the robot swarm within a certain distance from each other. Since GA searches for an optimal configuration with lower potential, the obtained paths are safer. The results of the simulation demonstrated that the proposed algorithm can plan the paths without collision.

This research work aims to provide a new solution to maritime pollution challenges by proposing a hybrid approach based on the two proposed solutions for trajectory planning ‘modified-GA’ and ‘modified-AA’.

3 Proposed approach

This paper presents a new hybrid approach based on the cooperation of air-sea vehicles. This cooperation minimizes the work overload to perform a monitoring and cleaning mission of dirty zones. The unmanned vehicles must plan their paths to achieve their goals in this mission. Thus, the approach proposed in [16] is developed by describing the architecture and constituent entities of the system, the trajectory planning methods, as well as their functioning.
Figure 1: Hybrid architecture of the system.

3.1 Architecture of the system

Figure 1 shows the hybrid architecture of the proposed system. It consists of a central unit, a monitoring vehicle to monitor maritime region, a swarm of cleaning vehicles to clean dirty zones, and a recovery vehicle to recover defective vehicles. The central unit is composed of a general coordinator, a base of life and a database. The coordinator general stores and consults the data in the database. It interacts with the base of life and the monitoring vehicle via a communication link of the IEEE.802.11a standard (5000m outside range) of the name Wi-Fi 5. While the Wi-Fi network of the IEEE.802.11b standard (35m-140m indoor range) allows messages to be connected between the monitoring vehicle and the swarm of cleaning vehicles.

3.2 Hierarchical role of each vehicle

The decision-making hierarchy of the proposed approach is made up at the first level of a general coordinator. It has the highest decision for the execution of various tasks and the launch and control of unmanned vehicles used, and that are located in the basis of life. The monitoring drone is loaded with a lower decision at a second level. It has the role to supervise and monitor its cleaning vehicle swarms. These cleaning vehicles are located in the third level and are composed by leader vehicles with their cleaning following vehicles. Their objective is to carry out the cleaning operation of the dirty zone according to the energy availability of each member. Each leader of the swarm has two necessary roles. It is responsible for the tasks/characteristics of its followers, and also, it shares (cooperates) with them the cleaning action. The coordinator launches a recovery vehicle to recover the failing vehicles.

3.3 Environment modeling

This section defines the modeling of this work environment along with the set of components. Table 1 summarizes the parameters used.

1. **Set of tasks**: five high level tasks are defined and used by the general coordinator, the monitoring and cleaning vehicles:
   - Allocating task \( t_a \) represents the task of allocating unmanned vehicles to different regions \( r \) and dirty zones \( z \);
   - Monitoring task \( t_{mr} \) represents a task of monitoring a region \( r \);
   - Cleaning task \( t_{cz} \) represents the task of cleaning a zone \( z \);
   - Supervising cleaning task \( t_{sz} \) represents the task of supervising the cleaning of a zone \( z \);
   - Launching task \( t_l \) represents the launching spot of the different previous tasks (monitoring, cleaning, supervising cleaning and allocating tasks).

2. **Set of vehicles**: different vehicles with their related roles are described as follows:
   - \( UAV_{mr} \): monitoring, homogeneous and semi-autonomous aerial vehicle; this vehicle incorporates a camera, an ultrasonic sensor, a GPS (Global Positioning System) and an autopilot software; it features a speed and an energy capacity that allows it to complete the following roles/tasks: monitor the regions, dirty zones, and supervise the cleaning vehicles swarm; ask / inform the task leader; launch / return the data and results to the general coordinator;
   - \( USV_{cz} \): cleaning, homogeneous and semi-autonomous surface vehicle; it has the same hardware and features as \( UAV_{mr} \) except it does not embed a camera; \( USV_{cz} \) is responsible for: cleaning the dirty zones; following / requesting tasks and informing / returning the data and results to its leader;
   - \( Leader_{cz} \): cleaning, homogeneous and semi-autonomous surface vehicle; this vehicle is similar to \( USV_{cz} \) in hardware and software components; it allows to: cooperate with followers (\( USV_{cz} \)) for cleaning mission; send the requests...
Table 1: Used parameters for the proposed approach.

| Parameters                                                                 | Description                                                                                                         |
|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| **Threshold**                                                             | Fixed threshold that allows classifying the zone coordinates according to the degrees of cells 'List_Degree_cz'.          |
| **Threshold_{average\_energy\_cap}**                                      | Threshold of the average energy capacity that has a fixed value and it enabled the determination of the average energy capacity. |
| **Threshold_{low\_energy\_cap}**                                           | Threshold of low energy capacity that has a fixed value and it enabled the determination of the low energy capacity.   |
| **List_{ID}(Id_{USV_{cz}})**                                              | List of USV_{cz} identifiers composing the swarm.                                                                     |
| **List_{energy\_usz(t)}(Id_{USV_{cz}}, Cap_{energy_{cz}})**               | List of the USV energy capacity in instant t during cleaning as well as its identifier.                              |
| **List_{zone}(List_{Degree_{cz}}, List_{Position_{cell}}, Position_{zone})** | List of coordinates of the dirty zone.                                                                               |
| **List_{zoneS}**                                                           | List_{zone} sorted by the degree of dirt.                                                                             |
| **List_{threshold\_degree_{cz}}**                                         | Coordinates list of the dirty zone that is compared with the threshold of dirt compared to the dirty degrees (List_{Degree_{cz}}). |
| **Nbr_{cz}**                                                              | A function that gives the number of selected USV_{cz} by General_{cz}                                                  |
| **Pos\_Start(x,y)**                                                      | Starting position that USV_{cz} can start from the base of life. This position is a Cartesian coordinate (x,y). Each USV_{cz} at its own starting position. |
| **Pos\_End(x,y)**                                                        | Final/End position represents the position of arrival of USV_{cz} to the dirty zone. This position is a Cartesian coordinate (x,y). Each USV_{cz} to its own ending position. |
| **Pos\_Goal(x,y)**                                                       | Goal position means the next position of USV_{cz} in its move. This position is a Cartesian coordinate (x,y).           |
| **List_{Pos}(x,y)**                                                      | List of Cartesian coordinates (x,y) representing the final positions (to arrive at the dirty zone).                   |
| **List\_taboo\_pos(Id_{USV}, Pos\_Goal(x,y))**                            | List to save positions that are already traversed by USV_{cz}. It contains the identifier of USV_{cz} which has already visited the position Pos\_Goal(x,y) where its Cartesian coordinate is the pair (x,y) in the grid. |
| **List\_taboo\_cell (Id_{USV}, Cell(x,y))**                              | List to memorize cells that are already cleaned by USV_{cz}. It is constituted by the identifier of USV_{cz} and its cleaned Cell(x,y). This cell is represented by the Cartesian couple (x,y) in the grid. |
| **List\_dist\_Cost (link_{ij}, Cost_{ij})**                              | List of distances between positions that are marked as an energy cost. It is composed of the link between the position i and j thus its energy Cost_{ij}. |
| **Energy_{cap_{cz}}**                                                    | Energy capacity of USV_{cz}.                                                                                          |
| **List\_virtual\_pos(List_{PosS}(x,y), Pos\_End(x,y))**                  | Virtual list which is constructed by a starting position list List_{PosS}(x,y) and Pos\_End(x,y).                      |
| **Parameters\_start\_up_{Md}(Id_{region}, Position_{region}, Path_{region})** | Triplet of start parameters for each UAV_{mr} which represents the identifier, position, and trajectory of a region. |
| **Parameters\_start\_up_{cz}(Id_{region}, Id_{zone}, Id_{Supmr}, Id_{Leader_{cz}}, Position_{zone}, Pos\_start(x,y), Pos\_end(x,y), List_{pos\_Ex(y)})** | Triplet of the start parameters for each USV_{cz} which represents the identifier of its region, zone, supervisor, leader and the position of zone with Pos\_Start(x,y) and Pos\_End(x,y) and List_{pos\_Ex(y)}. |
| **Parameters\_cleaning (Pos_{USV_{g}}, Cell_{cz}, Id_{region}, Id_{zone}, Position_{zone}, Trajectory_{zone}, Id_{Supmr}, Id_{Leader_{cz}})** | Cleanup parameters for USV_{cz} that replaces USV_{cz}(discharge). This list contains: the position of USV_{cz} in the grid, the cell to be cleaned, the identifier, the position and the trajectory of zone, the identifier of region, Supmr and Leader_{cz}. |
| **List\_characteristics (Id_{zone}, Id_{USV_{cz}}, Dur\_{ED1}, Cons\_{ED1}, Dur\_{ED2}, Cons\_{ED2}, Dur_{EC}, Cons_{EC})** | List of USV_{cz} characteristics where it contains the identifier of a zone and a USV_{cz}, the duration and the energy consumption of displacement of the base of life towards the dirty zone (Dur_{ED1}, Cons_{ED1}), the duration and the energy consumption of displacement in the dirty zone (Dur_{ED2}, Cons_{ED2}) and the duration and the energy consumption of cleaning (Dur_{EC}, Cons_{EC}). |
3. **Set of agents:** a description is provided for each used agent:

- **General**\textsubscript{crd}: General coordinator is represented by a laptop that contains a coordination software, guided by a human operator; it is responsible for the following: the base of life, data of the regions and the dirty zones; use (treatment)/storage of data in the database; launch of the tasks / missions and allocate tasks to vehicles;
- **Leader**\textsubscript{cz}: Leader vehicle is a USV\textsubscript{cz} that has two roles: it is an intermediary between the supervisor and USV\textsubscript{cz} swarm, and also cooperates in the cleanup operation;
- **USV\textsubscript{cz}(discharge):** Cleaning vehicle in discharge state is a cleaning vehicle, that has not yet completed its task and energy capacity are low during cleaning;
- **USV\textsubscript{cz}(free):** Cleaning vehicle in free state is a cleaning vehicle that has completed its task;
- **USV\textsubscript{cz}(prepared):** Cleaning vehicle in prepared state is a cleaning vehicle prepared in the base of life by the General\textsubscript{crd} which will replace USV\textsubscript{cz}(discharge);
- **Sup\textsubscript{mr} (USV\textsubscript{cz_discharge}):** Supervisor of USV\textsubscript{cz} (discharge) is the surveillance vehicle (USV\textsubscript{cz}) of the cleaning vehicle in discharge state;
- **Sup\textsubscript{mr} (USV\textsubscript{cz_free}):** Supervisor of USV\textsubscript{cz} (free) is the surveillance vehicle (USV\textsubscript{cz}) of the cleaning vehicle in free state;
- **Sup\textsubscript{mr} (USV\textsubscript{cz_prepared}):** Supervisor of USV\textsubscript{cz} (prepared) is the surveillance vehicle (USV\textsubscript{cz}) of the cleaning vehicle in prepared state;
- **Leader\textsubscript{cz}(USV\textsubscript{cz_discharge}):** Leader of USV\textsubscript{cz} (discharge) is the leader of the cleaning vehicle in discharge state;
- **Leader\textsubscript{cz}(USV\textsubscript{cz_free}):** Leader of USV\textsubscript{cz} (free) is the leader of the cleaning vehicle in free state;
- **Leader\textsubscript{cz}(USV\textsubscript{cz_prepared}):** Leader of USV\textsubscript{cz} (prepared) is the leader of the cleaning vehicle in prepared state.

4. **Set of regions:** the monitored maritime space is divided into maritime regions; the region is composed of two sub-spaces: an atmosphere sub-space where \textit{UAV\textsubscript{mr}} can be found, and a maritime sub-space with the swarm of USV\textsubscript{cz}, \textit{Vehicle\textsubscript{rec}}, base of life and dirty zones.

5. **Set of base of life:** it is a zone (can be a boat, ship or an island) to store a fixed number of \textit{UAV\textsubscript{mr}}, USV\textsubscript{cz} and \textit{Vehicle\textsubscript{rec}}.

6. **Set of database:** these are basics for storing and saving all the data and features of the maritime space as well as the different unmanned vehicles used.

7. **Set of dirty zones:** with water pollution, for example oil slicks or plastic waste; the proposed metric in this work is the degrees of dirt for each zone with four types: strong dirt, average-strong dirt, average dirt and low dirt; each zone is characterized by a list ‘List\textsubscript{zone}’ which delimits its borders by the coordinates; they are composed by the list the degrees of dirt ‘List\textsubscript{Degree\textsubscript{cell}}’, list of cell positions of this degrees ‘List\textsubscript{Position\textsubscript{cell}}’ and they are attached to a zone by ‘Position\textsubscript{zone}’.

### 3.4 Hybrid approach with two trajectory planning solutions

This section describes the key steps of the proposed hybrid approach with a proposal for tailored solutions. The first solution is based on a genetic algorithm (GA), and the second is based on an ant algorithm (AA). To this end, the approach is based on two essential steps: monitoring and cleaning. Both solutions are applied in the cleaning step. Figure 2 illustrates a structure of the proposed approach.

### 3.5 Step 1: Monitoring

This step presents the actions of the \textit{UAV\textsubscript{mr}} monitoring phases:

#### 3.5.1 Phase 1: monitoring setup

The monitoring step is performed, for example, twice a week in the maritime region to be monitored. The general coordinator General\textsubscript{crd} prepares the monitoring drones according to the region numbers, allocating a \textit{UAV\textsubscript{mr}} for each region. In addition, General\textsubscript{crd} checks periodi-
cally the maritime space statistics that are saved in the
database. If it finds a statistic region that contains a high
percentage of dirty zones in a certain period, it launches a
UAV_{mr} for a full day to monitor this dirty zone.

3. Once the maritime level data collection is complete,
UAV_{mr} sorts the collected list and saves it in a
List_{zones}. Then, it compares the List_{Degree} of
List_{zones} with a fixed threshold. After the comparison,
saves the result in List_{threshold_degreeZ} and sends it to
General_{crd}.

3.6 Step 2: Cleaning

This step is organized in three phases: cleaning setup, ex-
ecution and cleaning finalization.

3.6.1 Phase 1: cleaning setup

The two proposed solutions (modified-GA and modified-
AA) are applied exactly in this phase. In addition, they
guide the USV_{cz} swarm to execute the moving phase to
dirty zones. Before applying the two solutions, General_{crd}
analyzes the received data (List_{threshold_degreeZ}) from
each UAV_{mr}. Then, it determines the USV_{cz} number con-
taining the swarm for each dirty zone. This action is de-
scribed by Algorithm 1. Then, the main phases of the two proposed solutions are presented so that the USV$_cz$ swarm can plan its way to its dirty zone.

### 3.6.1.1 Solution 1: modified-GA trajectory planning

This phase is presented as the first solution modified to the trajectory planning problem. Different actions of this solution are mentioned so that the USV$_cz$ swarm selected can build its trajectory towards its dirty zone.

1. **Genetic algorithm setup:** the proposed GA is inspired by the algorithm presented in [24]. In this GA, the USV$_cz$ swarm must search for an optimal trajectory between the starting position (the base of life) and the goal (the dirty zone). Table 2 presents a comparative table of the proposed approach with the work of [24] in [16].

   - **Environment modeling:** the environment modeling depends on the map of the discretized region (metric map) by Sup$_mr$. The workspace is discretized by a square grid G which is composed of square cells. A node is located inside each cell to facilitate the movement of USV$_cz$. These nodes build a dynamic graph where the arcs are presented by $R_{ij}$ connection links. $R_{ij}$ between two neighboring cell-nodes represents the distance between two positions (Figure 3). This distance is represented by the energy value $E_{ij}$ that USV$_cz$ can consume in the displacement. Each cell-node $i_{th}$ represents a free / occupied position by a position or a non-solid obstacle in the environment. These positions are identified by Cartesian coordinates $(x,y)$ in a 2D plane. These positions have a maximum of eight links ($R_{ij}$) with the neighboring positions $j_{th}$.

   - **Encoding and generation of initial population:** after the environment modeling is finished, each node has its own number. The gene is made of one part which represents the number nodes. This gene corresponds to a position of USV$_cz$. The chromosome represents a mini-trajectory which contains a number of genes, the latter should belong to the rayon of the sensors. The population has a fixed length of individuals and each individual has a sequence of initial positions (cell-node). The mini-
trajectory planning is random by choosing neighboring positions in the known environment. As a result, individuals have their sequences of positions (mini trajectory), where a mini trajectory is chosen to execute it by USVcz (more clarification found in [16]).

- **Fitness Function and Selection:** fitness function is necessary to know the details and solution of the problem. Two parameters are taken into consideration to evaluate the fitness of the mini-trajectories: the total distance of mini-trajectory and the direction cost for each USVcz. An appropriate fitness function of mini-trajectory (i) is constructed as follows:

\[
F_i = A \cdot Dist_{\text{Trajectory}}(i) + B \cdot \sum Cost_{\text{Dir}}(i)
\]  

(1)

\[
Cost_{\text{Dir}}(i) = \text{Direct cost} + \text{Left cost} + \text{Right cost}
\]  

(2)

where, A and B are two equilibrium parameters (constant numbers) and \( Dist_{\text{Trajectory}}(i) \) and \( Cost_{\text{Dir}}(i) \) are the distance of the mini-trajectory (i) and the direction cost for each USVcz respectively. \( Dist_{\text{Trajectory}}(i) \) is calculated by the sum of each distance between two cell-nodes in which this distance is represented by a value of energy. \( Cost_{\text{Dir}}(i) \) is calculated according to the direction of USVcz when it moves to another node. It has three directions: direct direction (\( \text{Direct cost} \)) where USVcz moves to another position in the same direction; right/left direction (\( \text{Right cost}/\text{Left cost} \)) where it turns to the right/left of its position to get to another position.

- **Genetic operators:** new position sequences (new individuals) are created by applying crossover and mutation operators. Crossover is a performed positions sequence provided by two parents. These parents (two mini-trajectories) are selected by the tournament, where they (the individuals) are chosen at random and who with a low fitness value becomes a parent. A one-point crossover technique was chosen for this approach; two children are then added to the population. Figure 5 shows an example of a crossover operator, where the crossover point is the middle of the trajectory of parent 1. Then, child 1 takes the first part of the parent 1 chromosomes plus the second part of the parent 2 from the crossover point. Thus, child 2 takes the first part of parent 2 and the second part of parent 1. Therefore, two new mini-trajectories are provided. The mutation defines a quantity of genes that build a problem in the trajectory. The gene is replaced by one of the neighbor genes of its previous gene that are also randomly selected. Figure 6 shows the mutation operation on child 1 of the previous example. At each generation, applied elitism can be created by new individuals (children) of length 1 who have a low fitness to replace previous individuals (parents). Each new position sequence is executed by an associated USVcz, so that the fitness value of each new individual is calculated.

2. **Modified-GA based trigger process:** when \( \text{General}_{crd} \) chooses USVcz swarm, it sends the \( \text{List}_{\text{idusv}}(\text{Id}_{\text{USVcz}}) \) to their \( \text{Sup}_{mr} \). In return, \( \text{Sup}_{mr} \) sends its discrete maritime exploration map. \( \text{General}_{crd} \) broadcasts a set of parameters to the selected swarm, as shown in the sequence diagram of Figure 7.

After, \( \text{General}_{crd} \) launches the execution of GA and each USVcz of the swarm plans to move from the above mentioned steps of GA to the dirty zone.
3.6.1.2 Solution 2: modified-AA trajectory planning

This phase is presented as the second suitable solution for the trajectory planning problem.

Different actions of this solution are mentioned so that the USV_{cz} swarm selected can build its trajectory towards its dirty zone.

1. Ant algorithm setup: this solution is based on an ant algorithm proposed in [6, 25]. These algorithms show the steps of modeling ants’ behavior in the environment. To this end, the proposed modified-AA can model the behavior of a USV_{cz} swarm to find an optimal trajectory between the base of life and the dirty zone. In addition, the discretized grid G is composed of cells that contain positions. These positions are presented as cities. Each city \( i_{th} \) represents a free / occupied position by a USV_{cz} agent. These cities are identified by Cartesian coordinates (x,y) in a 2D plane. Cities \( i_{th} \) can have connection links (edges) \( R_{ij} \) between neighboring positions \( j_{th} \). \( R_{ij} \) represents the distance between cities (Figure 8). This distance is represented by an energy cost \( E_{ij} \) that a USV_{cz} agent will consume in the displacement between the cities.

![Figure 8: Environment modeling of the modified-AA.](image)

Based on the proposed algorithm, named ‘Modified-Ant Algorithm’, each agent USV_{cz} is placed in a starting city. USV_{cz} moves from one city to another using the connection links between cities towards the goal. \( u_i(t) \) is the vehicle number in city \( i \) at time \( t \) and \( U = \sum_{i=0}^{m} u_i(t) \) the total number of vehicles. The cities have three positions known by the USV_{cz}: starting city \( Pos_{Start(x,y)} \), goal city \( Pos_{Goal(x,y)} \) and final city \( Pos_{End(x,y)} \). If the end city \( Pos_{End(x,y)} \) equals the goal position \( Pos_{Goal(x,y)} \), then the USV_{cz} has arrived at the dirty zone. Each USV_{cz} agent has the following characteristics:

- It deposits a pheromone trace on \( R_{ij} \) when it moves from city \( i \) to city \( j \);
- It chooses the destination city according to a probability that depends on the distance between this city and its position and the amount of pheromones present on the edge (transition rule);
- To pass only once through each city, USV_{cz} cannot go to a city that has already been crossed, that’s why USV_{cz} must have a memory ‘List tabooPos(\( i_{th} \), \( Pos_{Goal(x,y)} \))’.

The traces of pheromones are modeled by the variables \( t_{ij}(t) \) which give the intensity of the trace on the trajectory \((i, j)\) at time \( t \). The transition probability from city \( i \) to city \( j \) by the agent \( u_i(t) \) is given by:

\[
P_{ij} = \begin{cases} \frac{[t_{ij}(t)]^{\alpha} [V_{ij}(t)]^{\beta}}{\sum_{l \in L_{u_i}} [t_{il}(t)]^{\alpha} [V_{il}(t)]^{\beta}}, & \text{if } j \notin L_{u_i}(i) \\ 0, & \text{else} \end{cases}
\]

where, \( L_{u_i}(i) \) represents the List tabooPos of \( u_i(t) \) located on the vertex \( i \) and \( V_{ij} \) represents a measure of visibility that corresponds to the inverse of the distance between cities \( i \) and \( j \). This list is represented by:

\[
V_{ij} \leftarrow \sum_{1}^{d_{ij}}
\]

where, \( d_{ij} \) is the distance between the city \( i \) and \( j \), which is presented by the \( E_{ij} \) in the array List_Cost. Then \( \alpha \) and \( \beta \) are two parameters for modulating the relative importance of pheromones and visibility. The update of the pheromones is done once all the ants have passed through all the cities:

\[
\tau_{ij} \leftarrow (1 - \rho) \tau_{ij} + \sum_{u=1}^{m} \Delta \tau_{ij}^u
\]

where \( \rho \) is a coefficient representing the evaporation of traces of pheromones. Then \( \Delta \tau_{ij}^u \) represents the link
reinforcement \((i, j)\) for \(u_i(t)\):

\[
\Delta r_{ij}^u = \begin{cases} 
\frac{Q}{L_i}, & \text{if USV passed through the arc } (i, j) \\
0, & \text{else} 
\end{cases}
\]

(6)

where \(Q\) is a constant and \(L_i\) is the length of the trajectory traveled by \(u_i(t)\) (the sum of the energy costs consumed during the travel between cities).

2. **Running of ‘Modified-AA’ algorithm:**

General\(_{crd}\) prepares a virtual list of \(\text{List\_virtualPos}(\text{List\_Pos}(x,y), \text{Pos\_End}(x,y))\). This list is constructed by a starting position list \(\text{List\_Pos}(x,y)\) and an ending position \(\text{Pos\_End}(x,y)\) (the arrival at the dirty zone). General\(_{crd}\) assigns a \(\text{Pos\_Start}(x,y)\) for each agent \(\text{USV}_{cz}\) of the selected swarm, and the \(\text{Pos\_End}(x,y)\) is shared in this swarm. General\(_{crd}\) launches Algorithm 2 to find the best trajectory that will apply to the swarm. The flow is started by initializing the transition trace \(\tau_{ij}(t)\) (3). An interval of time is proposed to assess the improvement of the trajectories built. Each \(u_i(t)\) will build its trajectory \(\text{Path}^u(t)\) from its \(\text{Pos\_Start}(x,y)\) to the \(\text{Pos\_End}(x,y)\).

This construction is made based on the transition rule (3). The agent \(u_i(t)\) detects its detection region via its ultrasonic sensor to identify its neighboring cities. From the maximum transition probability \(P_{ij}(t)\) that was computed, its next destination city is known. When the trajectory is complete, its length \(L^u_t\) is calculated. This trajectory is saved in a \(\text{List\_Path\_UT}\) list. At each end of the time lapse, the \(\tau_{ij}(t)\) (rule (5)) is updated. After, the best trajectory found in \(\text{List\_Path\_UT}\) is selected based on the maximum value of \(P_{ij}(t)\). The best trajectory of \(t\) is saved in a list \(\text{Path\_Final\_UT}\). When \(t = t_{max}\), the best trajectory \(\text{Best\_path}^u\) is selected of all \(t\) turns among the trajectories found in \(\text{Path\_Final\_UT}\). This trajectory is selected based on its maximum value of \(P_{ij}(t)\).

3. **Modified-AA based trigger process:** when General\(_{crd}\) chooses the \(\text{USV}_{cz}\) swarm, it sends the \(\text{List\_id\_sv}(\text{Id\_USV}_{cz})\) to its \(\text{Sup\_mr}\). In return, \(\text{Sup\_mr}\) sends its discrete maritime exploration map. General\(_{crd}\) broadcasts a set of parameters to the selected swarm, as shown in the sequence diagram of Figure 9. After, General\(_{crd}\) sends \(\text{Best\_path}^u\) to the swarm after changing the starting position of each \(\text{USV}_{cz}\). The latter moves together in the grid following the \(\text{Best\_path}^u\). Then, the first \(\text{USV}_{cz}\) is started from the \(\text{Pos\_Start}(x,y)\) of the trajectory, then the second starts from the first position of the first \(\text{USV}_{cz}\), and so on. Each \(\text{USV}_{cz}\) follows its neighbor \(\text{USV}_{cz}\).

The first \(\text{USV}_{cz}\) arrives at \(\text{Pos\_End}(x,y)\), every \(\text{USV}_{cz}\) except the first one checks its position to see if it is one of the positions found in \(\text{List\_Pos}(x,y)\) except the \(\text{Pos\_End}(x,y)\) of the first \(\text{USV}_{cz}\). Otherwise, each of these \(\text{USV}_{cz}\) performs a small function to arrive at one of the end positions (Algorithm 3). When all \(\text{USV}_{cz}\) are found in their end positions, General\(_{crd}\) changes the \(\text{Pos\_End}(x,y)\) of each \(\text{USV}_{cz}\) in the startup parameters.

### 3.6.2 Phase 2: cleaning operation

This phase allows the \(\text{USV}_{cz}\) swarm to move into the dirty zone and clean it. The maritime space is discretized in square grid (G). Each box of G can contain an object (a dirty cell / a clean cell). Objects are points in a numerical space with M dimensions; these objects to be partitioned are positioned. \(\text{USV}_{cz}\) can move in G and perceive a detection region \(R_s\) in their neighborhood (Figure 4). These \(\text{USV}_{cz}\) can...
clean the dirty cell type objects. To this end, Algorithm 4 applies the proposed approach with the two solutions. Using this algorithm, the swarm can move in the dirty zone and clean its dirty objects. This phase is started when the swarm arrives at its dirty zone. General\textsubscript{crd} stops the execution of modified-GA and modified-AA. Subsequently, General\textsubscript{crd} selects a Leader\textsubscript{cz} from the USV\textsubscript{cz} composing the swarm, and at the same time shares this information with the other USV\textsubscript{cz}. This leader has a very high energy capacity compared to other USV\textsubscript{cz}. After, General\textsubscript{crd} launches Algorithm 4 on the swarm and at the same time shares this information with the other USV\textsubscript{cz}. Leader\textsubscript{cz} launches the recording of the characteristics of himself and its followers. The USV\textsubscript{cz} of the swarm follow the algorithm so that they can properly position and move between the clean cells, select the cells to be cleaned afterwards, and share their positions between them.

### 3.6.3 Phase 3: cleaning finalization

This phase identifies the end of the cleaning step and collects the current characteristics of USV\textsubscript{cz} for each proposed solution, and has three cases:

1. **Case 1:** The cleaning task is completed. When the USV\textsubscript{cz} finishes its cleaning task, it returns to the base of life with a message of acceptance received by its Leader\textsubscript{cz} (further details in [16]).

2. **Case 2:** The cleaning task is not complete and Energy\textsubscript{capCZ} is average. In this case, two situations are available:
   - The first situation: If Energy\textsubscript{capCZ} of USV\textsubscript{cz} is greater than a Threshold\textsubscript{average_energy_cap}, i.e. its

   - energy capacity is average. So, its Sup\textsubscript{mr} finds a USV\textsubscript{cz} (free) in the swarm of the same zone (further details in [16]).
   - The second situation shows the case where there is not a USV\textsubscript{cz} (free) in the same zone. So, USV\textsubscript{cz} (discharge) informs its Leader\textsubscript{cz} by its List – energy\textsubscript{usv} (Id\textsubscript{USVcz}, Energy\textsubscript{capCZ}). Leader\textsubscript{cz}(USV\textsubscript{cz} discharge) sends List – energy\textsubscript{usv} and Parameters\textsubscript{cleaning} (Pos\textsubscript{USV}, Cell\textsubscript{cz}, Id\textsubscript{region}, Id\textsubscript{zone}, Position\textsubscript{zone}, Trajectory\textsubscript{zone}, Id\textsubscript{UAVMR}) to its Sup\textsubscript{mr}. After a waiting deadline, if Sup\textsubscript{mr} does not receive another message (a detection message of USV\textsubscript{cz} (free) in the other zone in the same region by Leader\textsubscript{cz}(USV\textsubscript{cz} free)) before exceeding this deadline, then it passes to the third situation. Otherwise, Leader\textsubscript{cz}(USV\textsubscript{cz} free) sends List\textsubscript{characteristics\textsubscript{usv}} with List – energy\textsubscript{usv} of
their USV<sub>cZ</sub>(free) to Sup<sub>mr</sub>. Sup<sub>mr</sub> processes the received message and selects USV<sub>cZ</sub>(free) that has a higher energy capacity. So, if it does not find any competent USV<sub>cZ</sub> then it passes to the third situation. Otherwise, Sup<sub>mr</sub> informs its Leader<sub>cZ</sub>(USV<sub>cZ</sub> free) by USV<sub>cZ</sub>(free) with the complete cleanup parameters (Sup<sub>mr</sub> changes the new position of USV<sub>cZ</sub>(free), I<sub>zone</sub> and new cell that USV<sub>cZ</sub>(free) will clean it with the same parameters I<sub>region</sub>, Position<sub>zone</sub>, Trajectory<sub>zone</sub>, Id<sub>USV<sub>mr</sub></sub>). Leader<sub>cZ</sub>(USV<sub>cZ</sub> free) returns the received message to its USV<sub>cZ</sub>(free). Leader<sub>cZ</sub>(USV<sub>cZ</sub> free) waits until the power capacity of USV<sub>cZ</sub>(discharge) becomes low to launch the USV<sub>cZ</sub>(free).

- The third situation is illustrated by Algorithm 5. This situation shows the case where Sup<sub>mr</sub> finds a USV<sub>cZ</sub>(free) that is assigned to dirty zones in the other region.

- The fourth situation is presented where there is not a free and competent USV<sub>cZ</sub> to complete the task of USV<sub>cZ</sub> in discharge state. So, Sup<sub>mr</sub>(USV<sub>cZ</sub> discharge) sends a message to the General<sub>crd</sub> to inform / require it of the need for a USV<sub>cZ</sub> with Parameters<sub>cleaning</sub>. It prepares a USV<sub>cZ</sub> in the base of life with the filled Parameters<sub>cleaning</sub>. After, it sends the identifier of USV<sub>cZ</sub>(prepared) with the cleaning parameters to Sup<sub>mr</sub> (USV<sub>cZ</sub> discharge). The latter sends this identifier to Leader<sub>cZ</sub>(USV<sub>cZ</sub> discharge) expecting the message of the energy capacity of USV<sub>cZ</sub>(discharge) becomes low.

3. **Case 3:** The cleaning task is not complete and Energy<sub>capCZ</sub> is low. This case is realized when the energy of USV<sub>cZ</sub>(discharge) becomes low; that is, the energetic capacity (Energy<sub>capCZ</sub>) of USV<sub>cZ</sub> is greater than a Threshold_low_energy<sub>cap</sub> (following the situations mentioned above). USV<sub>cZ</sub>(discharge) sends this information to its supervisor through its leader (all situations except the first situation: USV<sub>cZ</sub>(discharge) sends this information right to its leader and the leader informs it to return to the base of life). Then, Sup<sub>mr</sub>(USV<sub>cZ</sub> discharge) sends a message (Inform launch):

- **Directly to Leader<sub>cZ</sub>(USV<sub>cZ</sub> free) to USV<sub>cZ</sub>(free):** if this vehicle is owned in another zone in the same region (second situation);

- **General<sub>crd</sub> to another Sup<sub>mr</sub> (USV<sub>cZ</sub> free):** run the USV<sub>cZ</sub>(free) and prepare it with the cleaning parameters to calculate its trajectory and swim to its new dirty zone and clean it; General<sub>crd</sub> broadcasts this information to other Sup<sub>mr</sub> (third situation);

- **General<sub>crd</sub> to USV<sub>cZ</sub>(prepared):** launch USV<sub>cZ</sub>(prepared) which is in the base of life and to prepare it with the cleaning parameters; it can calculate its trajectory and swim to its new dirty zone and clean the remaining cells (fourth situation); Sup<sub>mr</sub>(USV<sub>cZ</sub> discharge) informs USV<sub>cZ</sub>(discharge) through its Leader<sub>cZ</sub>(USV<sub>cZ</sub> discharge) to return so that it can swim from its zone to the base of life (and to charge) (all situations except the first situation). Leader<sub>cZ</sub>(USV<sub>cZ</sub> discharge) sends List<sub>characteristics_usvs</sub> of USV<sub>cZ</sub>(discharge) to its supervisor and to General<sub>crd</sub> for

---

**Algorithm 5:** Third Situation (3).
the record. \( \text{Sup}_{mr}(\text{USV}_{cz\text{-discharge}}) \) informs the \( \text{Leader}_{cz}(\text{USV}_{cz\text{-free}}) \) to modify \( \text{List}_{\text{characteristics usvs}} \) (second and third situations) or prepare it for USVcz(prepared) (fourth situation). Finally, \( \text{Leader}_{cz}(\text{USV}_{cz\text{-free}}) \) informs its \( \text{USV}_{cz\text{(free)}} \) to start the cleanup and modifies at the same time the \( \text{List}_{\text{characteristics usvs}} \) of \( \text{USV}_{cz\text{(free)}} \) (first situation).

4 Logical formalization of the proposal

4.1 Conceptual model for planning

A conceptual model is a simple theoretical device to describe the main elements of a problem [26]. Most of the planning approaches described in [27] rely on a general model, which is common to other areas of computer science, namely the model of state-transition systems (also called discrete-event systems) [16, 26, 27]. The technical words of this formalizations (further details in [16]) are given in a general way.

Example 1. This example shows the state transition systems defined for three domains: UAV - Monitoring (or UAV - M domain applied to \( \text{UAV}_{mr} \)), Swarm (USV) - Cleaning (or SUSV-C domain applied to \( \text{USV}_{cz} \)) and USV (Substitute) - Cleaning (or USVS-C domain applied to \( \text{USV}_{cz} \)). These domains are presented by the following:

- **‘UAV - Monitoring’ domain**: Figure 10 shows a state transition system for a region involving two locations, a dirty zone, a base of life (for example: a boat) and a \( \text{UAV}_{mr} \). The set of states is \{s0, s1, s2, s3\} and the set of actions is \{stay in base, flap put base, move1 \& start-monitor, move2 \& end-monitor, discover, undiscover\}.

- **‘Swarm (USV) - Cleaning’ domain**: The system of region (Figure 11) involves three locations, two dirty zones, a base of life, object of the crane type for picking up, putting down and releasing unnamed vehicles. The set of states \{s0, s1, s2, s3\} and the set of actions is \{take, put, start-clean, end-clean, move1, move2\}.

- **‘USV(substitute) - Cleaning’ domain**: The system presented in Figure 12 is similar to the previous system. However, the difference is the use of a replacement cleaning vehicle (in free / prepared state), which will replace a USV in discharge state. The set of states is \{s0, s1, s2, s3, s4, s5\} with \{takeD \& move2, putD \& putD, moveD1 \& start-clean, moveD2 \& endclean, moveD1 \& move2, moveD2 \& movel, movel, move2\}.

Figure 10: A state transition system for the UAV-M domain.

Figure 11: A state transition system for the SUSV-C domain.

Figure 12: A state transition system for the UAVS-C domain.
4.1.1 A running example of 'UAV-M', 'SUSV-C' & 'USVS-C'

The planning procedures and techniques are illustrated in an example of simple nontrivial execution that include three domains, namely UAV-M, SUSV-C, and USVS-C. These domains complement each other. An abstract version of these domains can be defined from ten complete sets of constant symbols [16], namely a set of: regions \(R\), locations for each region \(L_r\), base of life \(B_b\), dirty zones \(Z_d\), monitoring vehicle \((UAV_{mr})\), cleaning vehicle swarm \((USV_{cz})\), cleaning vehicle in discharge state \((USVD_{cz})\), cleaning vehicle in free state \((USVF_{cz})\), cleaning vehicle in prepared state \((USVP_{cz})\) and a set of cranes \((C_{nb})\).

Thus, the topology of these domains is noted using the instances of predicates [16]: adjacent\((E, E')\) where localization \(E = \{L_r, R\}\) is adjacent to \(E = \{L'_r, R'\}\), belong \((C, L)\) where crane \(C\) belongs to location \(L\) and belong \((B, R)\) where a base of life \(B = B_b\) belongs to region \(R = R_b\). Also, the current configuration of the domains is denoted using instances of the following predicates, which represents the relationships that change over time [16], namely at\((NAME, E)\) where \(NAME = NAME1[UAV_{mr}, USV_{cz}]\) and \(NAME2[USVF_{cz}, USVP_{cz}, USVD_{cz}, SUSV_{cz}]\), occupied\((L)\), start-clean \((NAME, Z)\), end-clean \((NAME, Z)\), start-monitor \((NAME, R)\), end-monitor \((NAME, R)\), replace \((NAME, NAME)\) notreplace \((NAME, NAME)\), discover \((NAME, Z)\), undiscover \((NAME, Z)\), supervise \((NAME, NAME)\), notsupervise \((NAME, NAME)\), holding \((C, NAME)\), stayinbase \((NAME, B)\), flapoutbase \((NAME, B)\). Possible actions for this domains [16] can be: start-clean \((NAME, Z)\), end-clean \((NAME, Z)\), start-monitor \((NAME, R)\), end-monitor \((NAME, R)\), discover \((NAME, R, Z)\), undiscover \((NAME, R, Z)\), stayinbase \((NAME, B)\), flapoutbase \((NAME, B)\), move \((NAME, E1, E2)\), moveD \((NAME, E1, E2)\), take \((NAME, C, R)\), takeD \((NAME, C, R)\). The action 'moveD' is executed in parallel with the action 'start-clean', 'end-clean' and 'move'. The actions 'takeD' and 'putD' are executed in parallel with the action 'move'.

4.1.2 Representations for classical planning

The classical planning problems are presented in three different ways [26], namely: i) Set theoretic representation, ii) Classical representation, iii) State-Variable representation. This work focuses on the second representation (classical) to apply its planning on the proposed approach.

Example 2. This example illustrates a classical representation of the scenario 'UAV - M domain' and 'SUSV-C domain combined with USVS-C domain' which are described in Example 1:

- **For the UAV-M domain:** a UAV-M planning domain is formulated with a drone \((UAV_{11})\), a region \((R_1)\), a base of life \((B_{11})\), three locations \((L_{11}, L_{21}, L_{31})\), six dirty zones \((Z_{11}, Z_{32}, Z_{33}, Z_{22}, Z_{32}, Z_{53})\) and five vehicles \((USV_{11}, USV_{31}, USV_{26}, USV_{46}, USV_{66})\). The set of constant symbols is given by \((UAV_{11}, R_1, L_{11}, L_{21}, L_{31}, B_{11}, USV_{11}, USV_{31}, USV_{26}, USV_{66})\). One of the states is the state 3 illustrated in Figure 13.

\[
S3 = \text{belong}(B_{11}, R_1), \text{flapoutbase}(UAV_{11}, B_{11}), \text{start-monitor}(UAV_{11}, R_1), \text{discover}(UAV_{11}, Z_{31}), \text{discover}(UAV_{11}, Z_{22}), \text{discover}(UAV_{11}, Z_{63}), \text{ undiscover}(UAV_{11}, Z_{32}), \text{ undiscover}(UAV_{11}, Z_{42}), \text{ undiscover}(UAV_{11}, Z_{53}), \text{ supervise}(UAV_{11}, E_{11}), \text{ supervise}(UAV_{11}, E_{31}), \text{adjacent}(L_{11}, L_{21}), \text{adjacent}(L_{21}, L_{11}), \text{adjacent}(L_{21}, L_{31}), \text{occupied}(L_{11}), \text{occupied}(L_{21}), \text{occupied}(L_{31}).
\]

Figure 13: UAV-M planning domain.

- **For the SUSV-C domain combined with USVS-C domain:** an SUSV-C domain is illustrated with the USVS-C domain including a region \((R_1)\), three locations \((L_{11}, L_{21}, L_{31})\), eleven cleaning vehicles \((USVR_{34}, USV_{22}, USV_{52}, USVR_{62}, USV_{74}, USV_{44}, USV_{25}, USV_{55}, USV_{85}, USV_{66}, USV_{76})\), a crane \((C_{11})\) and five dirty zones \((Z_{12}, Z_{22}, Z_{63}, Z_{22}, Z_{53})\). The set of constant symbols is \((R_1, L_{11}, L_{21}, L_{31}, USVR_{34}, USV_{22}, USV_{52}, USVR_{62}, USV_{74}, USV_{44}, USV_{25}, USV_{55}, USV_{85}, USV_{66}, USV_{76})\). Figure 14 illustrates a state of this domain.

\[
S3 = \text{belong}(C_{11}, L_{11}), \text{holding}(C_{11}, USVR_{34}), \text{end-clean}(USV_{52}, USV_{62}, USV_{22}, Z_{26}), \text{start-clean}(USV_{74}, USV_{44}, Z_{42}), \text{start-clean}(USV_{55}, USV_{25}, USV_{85}, Z_{53}), \text{start-clean}(USV_{66}, USV_{67}, Z_{63}), at(USV_{22}, USV_{52}, USV_{62}, L_{21}), \text{replace}(USVR_{34}, USVD_{44}), \text{replace}(USVR_{34}, USV_{44}), at(USV_{34}, L_{11}), \text{adjacent}(L_{11}, L_{21}), \text{adjacent}(L_{21}, L_{11}).
\]
A hybrid air-sea cooperative approach combined with a swarm trajectory planning method

adjacent($L_{21}, L_{31}$), adjacent($L_{31}, L_{22}$), occupied($L_{11}$), occupied($L_{21}$), occupied($L_{31}$).

Figure 14: SUSV-C planning domain combined with USVS-C planning domain.

5 Simulation

An illustrative example with two scenarios for dirty zones (with a homogeneous degree of dirt and a low degree of dirt) is provided to simulate the functioning of the proposed approach. Two trajectory planning solutions are given for swarm $USV_{cz}$, namely the modified-GA and the modified-AA. The following measures allow us to highlight the contributions of the proposed approach: displacement energy consumption between the base of life and dirty zone, displacement-cleaning energy consumption in the dirty zone and total energy consumption of swarm $USV_{cz}$.

5.1 Discretization of the virtual environment

Figure 15 illustrates a simplistic example of the virtual environment. The maritime space of a ‘Region’ is assumed to include a polluted ‘Zone’; a central unit consists of a base of life. Two different dirty zones are proposed according to the degrees of dirt in this region. After randomly defining the degrees of dirt in the environment with the two different zones, the discretization step is activated. Two matrices are obtained, where the black cells represent the dirty zone (Figure 16 and Figure 17).

5.2 Results

Two modified solutions for trajectory planning were used in this work. These solutions were implemented using a genetic algorithm and an ant algorithm. The two modified algorithms were programmed via open source Java. The simulations were run on a PC with a Core (TM) i5-5200U @ 2.20 GHz running the Windows 7 Professional operating system.

The proposed approach was implemented for a maritime region without hard obstacles that can prevent vehicles from navigating and plan their trajectories in this environment. This region includes a monitoring vehicle ($UAV_{mr}$), a cleaning vehicle swarm ($USV_{cz}$) and a dirty zone. Based on a metric map (Grid), the $UAV_{mr}$ captured data from its region. This data was measured against a color metric. These colors were partitioned on four intervals: $[0, 25], [25, 50], [50, 75], [75, 100]$ involve a white cell (weak dirt), light brown (average dirt), dark brown (medium-strong dirt) and black (strong dirt). The $UAV_{mr}$ classifies this data (degrees) of the cells by comparing the predefined threshold value (equal to 25% of the degree of dirt) with the Degree$_{cell}$ of each cell. The number of $USV_{cz}$ containing the swarm used in the simulations varied from 3 $USV_{cz}$ to 7 $USV_{cz}$. Each $USV_{cz}$ swarm planned its movement based on optimal trajectories to reach its dirty zone and clean it. These trajectories were built based on a modified-GA or a modified-AA. The modified-GA was executed successively on each $USV_{cz}$ of the swarm.

The population was composed of five individuals and the maximum number of generations equaled to five because the solution obtained from the fourth generation was identical to that of the fifth generation compared to a
tolerated threshold. The initial population was randomly generated where each individual (trajectory) had a start and end position. The tournament selection was applied to find the individual (trajectory) with a weak fitness (energy consumed). This fitness function used two balancing parameters: \( A = 0.02, B = 0.01 \). The crossover was used on this individual (the description cited in sub-section ‘Solution 1: modified-GA trajectory planning’). The mutation was not applied in this work because it was noticed that the two new children (trajectories) are 100% correct at the end of each generation.

In addition, the parameters of the modified-AA were used to find the best trajectory that will be applied to the swarm of \( USV_{cz} \) are: \( \tau_0 = 0.1, \rho = 0.5, a = 2, b = 1, n = 90 \); number of time laps between \( t_0 = 0 \) and \( t_{max} = 4 \). The cost value between the cells in the grid is randomly generated between 5 and 10. The energy required for the \( USV_{cz} \) of the swarm to turn left / right from its position to another position is 0.2% and 0.05% be able to continue directly. When the \( USV_{cz} \) swarm arrives at its zone, it starts cleaning dirty cells based on the proposed Algorithm 4. The proposed energy required to clean a black cell is 0.9%, for a medium-high dirt cell of 0.5% and for an average dirt cell of 0.2%. A formula was developed to calculate the total energy consumption (TEC) of each \( USV_{cz} \) swarm: 

\[ TEC = DEC1 + DCEC (DEC2 + CEC) \]

This formula groups the displacement energy consumption of the base of life to the dirty zone (DEC1), the displacement energy consumption (DEC2) plus the cleaning energy consumption (CEC) in the dirty zone (DCEC).

### 5.2.1 Scenario 1: Zone with a strong degrees of dirt

This first scenario shows the simulation results of a dirty zone with strong degrees of dirt. These results give the curves of DEC1 (or DEC), DCEC and TEC of \( USV_{cz} \) swarm for the two proposed solutions:

- **Result of displacement Energy Consumption (DEC1):** Figure 18 shows the results of the displacement energy consumption of each \( USV_{cz} \) swarm from the base of life to its dirty zone. The modified-GA was compared with the modified-AA in terms of energy consumption (DEC1). It can be noted that the DEC curve of the modified-AA is less than the DEC curve of the modified-GA with a minimum gain of 1.01% and a maximum gain of 3.51%. Therefore, \( USV_{cz} \) consumes less energy to plant its trajectory using the modified-AA algorithm compared to the modified-GA with an average gain of 2.62%.

- **Result of displacement-cleaning energy consumption (DCEC):** Figure 19 shows a simulation to evaluate the \( USV_{cz} \) swarm behavior by its energy consumption for displacement and cleaning its dirty zone. Table 2 shows that the DCEC of the modified-AA decreases when the swarm is composed of 3, 4, 6 and 7 \( USV_{cz} \) compared to the second DCEC with a low minimum gain of 0.5%. On the other hand, it increases rapidly when the swarm contains 5 \( USV_{cz} \) with a maximum gain of 1.20%. Thus, the modified-GA is generally better than the modified-AA to move and clean with a \( USV_{cz} \) swarm in a zone with strong degrees of dirt. Therefore, an average gain of 0.4% is obtained.

- **Result of total energy consumption (TEC):** the two previous results were combined to calculate the total energy consumption based on the two proposed algorithms. Figure 20 shows that the TEC curve of the modified-AA is below the TEC curve of the modified-GA with a minimum gain of 1.13% and a maximum gain of 3.5%. Therefore, the proposed modified-AA proposal consumes less energy than the modified GA with an average gain of 2.4% in a zone with a strong degrees of dirt.

| Numbers of \( USV_{cz} \) | CEC_GA | CEC_AA |
|---------------------------|-------|-------|
| 3                         | 292,25| 289,75|
| 4                         | 347,15| 344,65|
| 5                         | 331,15| 451,7 |
| 6                         | 353,9 | 354,2 |
| 7                         | 404,3 | 409,6 |
| **Average gain:**         |       |       |
| **Minimum gain:**         | 0.5%  |       |
| **Maximum gain:**         | 1.2%  |       |

### 5.2.2 Scenario 2: Zone with a low degree of dirt

The second scenario shows the simulation results for a zone with a low degrees of dirt. These results give the curves of DEC1 (or DEC), DCEC and TEC of \( USV_{cz} \) swarm for the two proposed solutions:

- **Result of displacement energy consumption (DEC1):** Figure 21 shows the results of DEC1 of each \( USV_{cz} \) swarm from the base of life to its dirty zone. The
modified-GA was compared with the modified-AA with respect to DEC1. It can be noted that the DEC1 curve of the modified-AA is less than the DEC1 curve of the modified-GA with a minimum gain of 0.63% and a maximum gain of 3.4%. Therefore, the USVcz swarm consumes less energy to plan its trajectory using the modified AA algorithm compared to the modified-GA with an average gain of 1.71%.

- **Result of displacement - cleaning energy consumption (DCEC):** Figure 22 presents a simulation that evaluated the USVcz swarm behavior in DCEC in a zone with a low degree of dirt. Table 3 shows that the swarms of 3, 4, 5 and 7 USVcz consume almost the same displacement energy and clean-up by applying the modified-AA and modified-GA with a minimum gain of 0.01%. In addition, the amount of energy consumed by a swarm of 5 USVcz using the modified-AA increases rapidly with a maximum gain of 0.4% compared to the modified-GA. Thus, the modified-GA is generally better than the modified AA in moving and cleaning for the USVcz swarms in this dirty zone with an average gain of 0.4%.

| Numbers of USVcz | CEC_AA | CEC_GA |
|------------------|--------|--------|
| 3                | 266.95 | 267.35 |
| 4                | 285.5  | 285.2  |
| 5                | 324.2  | 322.8  |
| 6                | 364.8  | 329.05 |
| 7                | 380.05 | 380.25 |
| Average gain:    | Minimum gain: | Maximum gain: |
| 0.4%             | 0.01%   | 0.4%    |

- **Result of total energy consumption (TEC):** The two previous results were used to calculate TEC based on the two proposed algorithms. Figure 23 shows that the TEC curve of the modified-AA is below the TEC curve of the modified-GA with a minimum gain of 0.63% and a maximum gain of 3.4%. Therefore, the proposed modified-AA consumes less energy than the modified-GA with an average gain of 1.64% in a zone with a low degree of dirt.

### 5.2.3 Discussions

The USVcz swarm trajectory planning applied in this approach is based on two modifiable meta-heuristic methods: modified-AA and modified-GA. The implementation of these algorithms shows that the modified-GA takes a lot of computation time in the execution of its steps compared to the modified-AA. Thus, the convergence of its best solution reached a number of strong iterations compared to the modified-AA. In addition, a single path obtained/generated by the modified-AA is applied to a USVcz swarm in a sequential manner to arrive at its zone. However, the modified-GA generates a trajectory for each USVcz of a swarm while respecting the shape of a swarm. Therefore, the results of the simulation show that the USVcz swarm consumes less displacement and cleaning energy using the modified-AA compared to the modified-GA in zones with different degrees of dirt.

### 5.2.4 Positioning of the approach

This section positions the proposed approach in relation to the related works cited in Section 2. Each study mentioned has the characteristics / parameters that differentiate it from the others. Table 4 shows a comparison study of these studies and the advantage of the proposed approach. In this hybrid approach, a method of cooperation and coordination between a General, a semi-autonomous UAV mr, and a USVcz swarm was proposed to accomplish the cleaning missions of dirty oceanic regions contrary to the work of [17]. In [17], the authors developed a complete framework for the design and control of swarms of autonomous collaborative robots, with particular emphasis on quadcopters collaborating with each other to perform space tasks.

Furthermore, the work in [13] proposed a trajectory planning guidance system based on a USVcz swarm. In addition, the article of [3] enhanced the first system for an ASVs, an AUV and a guide. The UAV mr of this approach locates and detects the dirty zone on the basis of its sensors like the developed USV [19], which enables the location of the gas source in the sea by two gas sensors. In [21], it extracts a sample of the surface water and runs an image-based detection algorithm to confirm the presence of oil. Depending on the data received and the map being explored by the UAV mr, the general coordinator can send a swarm of USVcz to each dirty zone by executing Algorithm 1.

Thus, in this paper two solutions were proposed for the trajectory planning from the base of life to the dirty
zone for each \( USV_{cz} \) swarm. The first solution is based on a modified-GA compared to the GA proposed in [24]. While the second modified solution is inspired by the ant algorithms proposed in [6, 25]. The swarm is moved in a square grid (space 2 D) discretized by \( UAV_{mr} (Sup_{mr}) \). The cells are marked based on the degrees of dirt that the \( UAV_{mr} \) processes according to the colors captured. On the other hand, the work of [5] used a grid that is modeled by black (obstacles) and white (clean) cells. In addition, a Voronoi free space scheme for modeling the work environment used in [12]. The swarm applies both solutions to find its trajectory with less distance. The fitness function of the modified-GA is calculated in relation to the energy consumed between the positions traveled and the energy consumed by all the actions performed (turn left / right or continue directly) by \( USV_{cz} \).

On the other hand, the fitness of [9, 12, 22] is based on the distance traveled and the Euclidean distance of the goal. The tournament selection method was applied in the modified AA while the roulette method was applied in [9, 11]. The \( USV_{cz} \) swarm trajectory planning problem was managed to find its shortest trajectory. During simulation, it was noticed that the proposed algorithm MMAS [11] offers very good performances for the optimal path compared to a GA and an algorithm A*. Thus, the GA of [9] has a higher average performance than the one proposed by A* and C*. For this purpose, a dirty zone cleaning algorithm was applied to the \( USV_{cz} \) swarm where each swarm can at the same time move and clean its dirty cells without specifying how to remove the dirt (the oil spill). On the other hand, the swarms of labor robots [20] can place the barge with oil suction equipment and move it to another location to remove the oil more safely.

The two proposed solutions were also compared with respect to the two different scenarios, namely a zone with a homogeneous degree of dirt and a zone with a strong degree of dirt. These scenarios were simulated according to three metrics: energy consumption of displacement of the base of life towards the dirty zone, energy consumption of displacement and cleaning in the dirty zone, and total energy consumption. The simulation results obtained confirmed the choice to use both solutions. Thus, the modified-AA algorithm gives encouraging results compared to the modified-GA algorithm.
Table 4: Comparison between several related works.

| Work / Criteria | [13] | [3] | [9] | [11] | [9] | [12] | [19] | [20] | [21] | [17] |
|-----------------|------|-----|-----|------|-----|------|------|------|------|------|
| System type     | Cooperative orientation system | USV | Swarm path-following guidance system | USV | Optimal Planning System-complex obstacles | Robot exploratory path planning system | Robot exploratory path planning System | Movement planning system-robot swarms | On-Board real-time in-flight trajectory planner (RTTP) | Prototype of a USV | New robotic system swarm | USV design system | Organic computing approach | Monitoring-cleaning system and USV swarms trajectory planning |
| Main objective  | Monitor/help human divers at sea-mission execution | Combine swarm behavior-following guidance system | Find shortest optimal path | Find an optimal path between start/goal position | Plan collision-free paths | Volcanic monitoring and ash sampling task | Navigation and localization the gas source of an oil spill | Oil Spill Cleaning Up | Oil Spill Handling | Collaboration of autonomous robot swarms to perform spatial tasks | Development a framework for observer/ controller of this robot swarms | Monitor ocean region UAV, cold security zone and plan trajectory-USV swarm |
| Used type of method | Path-following cooperative, method-virtual target | Swarms aggregation, module-virtual target | Intelligent route planning-moulette, grid-obstacles | Max-Min System algorithm | Genetic algorithm (GA) | Potential-based GA | Collisionless trajectory planning | Research the road to get to the gas source (navigation) | Sensing, localization, navigation and collection of oil spills | Detection and collection of samples in oil spills | Development of O/C architecture modules, behavioral and swarm control method | Modified-GA and modified-AA |
| Gear type | One ASV, one ROV/AUV | Four charlie USV | One USV | Pioneer P3AT Robot | One robot | 8 robots | One Fixed wing UAV (4 Flights) | One USV | Swarm of ship robots | One USV | One robot, two/four AR.Drone Parrot 2.0 quad-rotor | One UAV, Swarm USV |
| Environment | 3D marine complexes | Disrupt-marine | Obstacle environment | Previously unknown | Not cited | 3D workspaces | Volcán de Fuego in Guatemala | Sea | Sea | Sea | Working area (simulation) | Maritime-Atmosphere space |
| Algorithm / Functions | Function Lyapunov | Function Lyapunov | Ant Colony | MNAS, A*, GA | A*, GA | Repulsion-spring function, Voronoi and GA | RTTP based on GA | Gas sensors (concentration) and fuzzy logic controller | Navigation method and operation of the swarm, cleaning method | Straight-line tracking algorithm with GPS navigation, sampling mechanism and image-based detection algorithm (using automatic learning) | Methods of O/C architecture modules, behavioral and swarm control method | Monitor numbers, M-AA, position adjustment, cleaning and situation 3 |
| Selection types | Not cited | Not cited | Not cited | Not cited | Tournament | Highest percentage (10%) | Roulette | Not cited | Not cited | Not cited | Not cited | Tournament |
| Fitness | Not cited | Not cited | Not cited | Traveled-Euclidean distance | Not cited | Configuration formula (defined) | Less displacement energy | Low cost | Not cited | Not cited | Not cited | Not cited |
| Metrics to measure | Horizontal motion-speed of vehicles | Robots movements speed, guidance-following path | Optimize path-obstacle environment | Distance traveled by ants | Distance traveled by robots | Behavior of swarm trajectories in optimizing the configuration | Cost consumption of the road with and without obstacle | USV speed, gas sensing on the USV and error rate in the direction of the goal | Consumption of money and time (in collecting small oil spills) | Detection accuracy to classify samples | Adaptability of individual drone and optimization of the collaboration | DEL, CEC and TEC of USV | swarm |
| Simulator | HIL(10): ROS environment | Kinematic-dynamic simulation-model | MATLAB | Player-Stage Tools | P-S Tools | PC (Core2 Duo 1.7 GHz CPU, XP) | Hardware (Skywalker X8 airframe), Mission Control (MAVLink, ArduPilot, etc.), GCS | USV Prototype | Robot design (GPS module, communication System, monitoring program, etc) | Robot Operating System (ROS), CORE2-ROS controller | Robot Operating System (ROS), Gazebo and Unity3D environment | PC (Corei5-5200U-CPU,2.7 GHz/W7P)-java |
6 Conclusion

This article has presented a hierarchical decision-making system for a hybrid air-sea approach. This approach uses a UAV$_{mr}$ for each maritime region and a USV$_{cz}$ swarm to clean dirty zones. In the monitoring step, the color was chosen as a degree of dirt for UAV$_{mr}$ to detect the dirty zones and the start of cleaning. During the cleaning phase, two solutions were proposed: ‘modified-GA’ and ‘modified-AA’. To this end, a USV$_{cz}$ swarm plans its displacement from the base of life towards its dirty zone. When this swarm arrives at its dirty zone, it begins to move and clean its dirty cells based on the proposed Algorithm 4. The USV$_{cz}$ swarm measures its amount of energy based on an energy threshold, and then sends its information to its Leader$_{cz}$. The latter returns this information to its Sup$_{mr}$ to find a USV$_{cz}$ competent. Sup$_{mr}$ launches the USV$_{cz}$(competent) which will replace the USV$_{cz}$(discharge) when the energy amount of the latter becomes low. The proposition is formalized by means of a classical representation.

Two scenarios were proposed to simulate this proposal, namely a zone with a strong degree of dirt and a zone with a low degree of dirt. The measured metrics are the displacement energy consumption, the displacement energy consumption plus cleaning, and the total energy consumption for each USV$_{cz}$ swarm. The situation where the USV$_{cz}$ breaks down was not simulated in this work (a simulation was performed in [16, 23]). The simulation results show that the proposal applied with the modified-AA gives encouraging results compared to the modified-GA. This work did not address the problem of failure of the General$_{cz}$, Leader$_{cz}$ and the UAV$_{mr}$ in the execution of the tasks. Thus, the Leader$_{cz}$ was changed after the trajectory planning phase by one of its followers due to the lack of a powerful leader in terms of energy able to move, clean and receive / transmit data.

Therefore, as future work, the authors intend to develop a hybrid approach to address this problem. In addition, the authors aim to develop the modified-AA with a proposal of the wave propagation instead of pheromone, and to study the influence of speed variations of unmanned vehicles. Furthermore, the authors would intend to assign a cluster of drones for each region in a complex environment through solid obstacles, thus implementing an intelligent planning approach similar to the PSO method using fuzzy logic in combination with the bee colony algorithm.

References

[1] M. Kloetzer and C. Belala, “Temporal logic planning and control of robotic swarms by hierarchical abstractions,” in Proceedings of IEEE Transactions on Robotics (T-RO), vol. 23, no. 2, pp. 320–330, 2007.

[2] M. M. Al-Rifaie and J. M. Bishop, “Stochastic diffusion search review,” Paladyn, J. Behav. Robot., vol. 4, no. 3, pp. 155–173, 2013.

[3] M. Bibuli, G. Bruzzone, M. Caccia, A. Gasparri, A. Priolo, and E. Zereik, “Speed constraints handling in USV swarm path-following frameworks,” in IFAC Proceedings Volumes, vol. 46, no. 33, pp. 73–78, 2013, https://doi.org/10.3182/20130918-4-JP-3022.00031.

[4] S. Xiao-Wei, Study of mobile robot path planning, [D] Shenyang: Northeastern University, China, 2008.

[5] W. Yuan-Hui and C. Cen, “Research on optimal planning method of USV for complex obstacles,” in Proceedings of IEEE ICMA 2016, Harbin, 2016, pp. 2507–2511.

[6] N. Labroche, N. Monmarché, and G. Venturini, “Virtual clustering with artificial ants colonies,” in Proceedings of International Conference on KES-2003, Berlin, Heidelberg, 2003, pp. 332–338.

[7] R. Kala, A. Shukla, and R. Tiwari, “Robot path planning using dynamic programming with accelerating nodes,” Paladyn, J. Behav. Robot., vol. 3, no. 1, pp. 23–34, 2012.

[8] D. Ferguson and A. Stentz, “The field D* algorithm for improved path planning and replanning in uniform and non-uniform cost environments”, Robotics Institute, Carnegie Mellon University, Pittsburgh: Tech. Rep, 2005.

[9] V. de Carvalho Santos, C. F. M. Toledo, and F. S. Osório, “An exploratory path planning method based on genetic algorithm for autonomous mobile robots,” in Proceedings of 2015 IEEE CEC, Sendai, Japan, 2015, pp. 62–69.

[10] L. Gao, S. Pan, and J. Shen, “A robot path planning scheme based on neural network,” J. Theor. App. Info. Techn., vol. 46, pp. 654–658, 2012.

[11] V. D. C. Santos, F. S. Osório, C. F. Toledo, F. E. Otero, and C. G. Johnson, “Exploratory path planning using the max-min ant system algorithm,” in Proceedings of 2016 IEEE CEC, Vancouver, BC, Canada, 2016, pp. 4229–4235.

[12] C. C. Lin, K. Chen, P. Y. Hsiao, and W. J. Chuang, “Motion planning of swarm robots using potential based genetic algorithm,” Int. J. Innov. Comput., vol. 3, no. 1, pp. 23–34, 2012.

[13] M. Bibuli, G. Bruzzone, M. Caccia, A. Ranieri, and E. Zereik, “Multi-vehicle cooperative path following guidance system for diver operation support,” IFAC-PapersOnLine, vol. 48, no. 16, pp. 75–80, 2015.

[14] M. Bibuli, M. Caccia, and L. Lapierre, “Virtual target based coordinated path-following for multi-vehicle systems,” IFAC Proceedings Volumes, vol. 43, no. 20, pp. 336–341, 2010, https://doi.org/10.3182/20100915-3-DE-3008.00014.

[15] M. Bibuli, A. Gasparri, A. Priolo, G. Bruzzone, and M. Caccia, “Virtual target based path-following guidance system for cooperative USV swarms,” IFAC Proceedings, vol. 45, no. 27, pp. 362–367, 2012, https://doi.org/10.3182/20120919-3-IT-2046.00062.

[16] S. Bella, A. Belbachir, and G. Belalem, “A hybrid architecture for cooperative UAV and USV swarm vehicles,” in Proceedings
A hybrid air-sea cooperative approach combined with a swarm trajectory planning method

[17] S. Mammen, et al., “OCbotsics: an organic computing approach to collaborative robotic swarms,” in Proceedings of IEEE SIS’14, Orlando, Florida, U.S.A., 2014, pp. 1–8.

[18] A. Tahir, J. Böling, M.-H. Haghbayan, H. T. Toivonen, and J. Plosila, “Swarms of unmanned aerial vehicles – a survey,” J. Ind. Inf. Integration, vol. 16, 100106, 2019.

[19] M. L. R. Putra, M. Rivai, and A. N. Irfansyah, “Unmanned surface vehicle navigation based on gas sensors and fuzzy logic control to localize gas source,” J. Phys. Conf. Ser., IOP Publishing, IOP Publishing, vol. 1201, p. 012001, 2019.

[20] E. M. H. Zahugi, M. M. Shanta, and T. V. Prasad, “Oil spill cleaning up using swarm of robots,” Advances in Computing and Information Technology, Berlin, Heidelberg: Springer, 2013, pp. 215–224.

[21] W. Al Maawali, M. Al Naabi Al Yaruubi, A. Saleem, and A. Al Maashri, “Design and implementation of an unmanned surface vehicle for oil spill handling,” in Proceedings of IEEE 1st International Conference on UVS-Oman 2019, Muscat, Oman, 2019, pp. 1–6.

[22] B. Schellenberg, T. Richardson, A. Richards, R. Clarke, and M. Watson, “On-board real-time trajectory planning for fixed wing unmanned aerial vehicles in extreme environments,” Sensors (Basel), vol. 19, no. 19, p. 4085, September 2019.

[23] S. Bella, A. Belbachir, and G. Belalem, “A centralized autonomous system of cooperation for uavs- monitoring and USVS-cleaning,” Int. J. Soft. Innov., vol. 6, no. 2, pp. 50–76, 2018.

[24] M. A. Yakoubi and M. T Laskri, “The path planning of cleaner robot for coverage region using genetic algorithms,” Journal of Innovation in Digital Ecosystems (JIDES), vol. 3, no. 1, pp. 37–43, 2016, https://doi.org/10.1016/j.jides.2016.05.004.

[25] C. Guéret, N. Monmarché, and M. Slimane, “Ants can play music,” in International Workshop on ANTS 2004, Berlin, Heidelberg, 2004, pp. 310–317.

[26] M. Ghallab, D. Nau, and P. Traverso, Automated Planning: Theory and Practice, Edition Elsevier, 2004.

[27] M. Ghallab, D. Nau, and P. Traverso, Automated Planning and Acting, Cambridge University Press, 2016.