Distributed coverage optimization for a fleet of unmanned maritime systems for a maritime patrol and surveillance application

Geert De Cubber
dept. of Mechanics
Royal Military Academy
Brussels, Belgium
geert.de.cubber@rma.ac.be

Rihab Lahouli
dept. of Mathematics
Royal Military Academy
Brussels, Belgium
rihab.lahouli@mil.be

Daniela Doroftei
dept. of Mechanics
Royal Military Academy
Brussels, Belgium
daniela.doroftei@rma.ac.be

Rob Haelterman
dept. of Mathematics
Royal Military Academy
Brussels, Belgium
rob.haelterman@mil.be

***Abstract*—In order for unmanned maritime systems to provide added value for maritime law enforcement agencies, they have to be able to work together as a coordinated team for tasks such as area surveillance and patrolling. Therefore, this paper proposes a methodology that optimizes the coverage of a fleet of unmanned maritime systems, and thereby maximizes the chances of noticing threats. Unlike traditional approaches for maritime coverage optimization, which are also used for example in search and rescue operations when searching for victims at sea, this approaches takes into consideration the limited seaworthiness of small unmanned systems, as compared to traditional large ships, by incorporating the danger level in the design of the optimizer.**

***Index Terms*—unmanned maritime systems, maritime surveillance, distributed coverage optimization**

I. Introduction

An ever-increasing percentage of the world population is living in coastal areas. As a result, also more and more criminals are turning to our seas and oceans to carry out illegal activities, such as drugs smuggling, human trafficking, illegal fishery, etc. The problem for law enforcement agencies is that patrolling and surveilling all these vast ocean surfaces is impossible with traditional means from an economic and operational point of view.

Unmanned Maritime Systems (UMS) could provide maritime law enforcement agencies with a valuable tool for increasing their capabilities, certainly when they are incorporated in a much wider maritime situational awareness toolkit [1], encompassing also satellite monitoring [2], manned and unmanned aerial assets [3] with advanced analytics solutions that can turn the data gathered by all these agents into information and knowledge.

One of the main capabilities the UMS need to possess is the capability to operate together as a well-coordinated group or team, working together towards a higher-level goal such as maritime surveillance. However, the practical deployment of these novel smaller-scale UMS requires the careful consideration of several aspects related to the design of the surveillance architecture. As an example, the classical approaches towards distributed patrol and surveillance of maritime environments by manned systems do not take into consideration the effects of small waves (which are irrelevant for larger ships, but very important for small UMS).

In this paper, we will therefore propose a novel methodology for the real-time control of a fleet of two to ten UMS. The presented methodology is casted as a distributed coverage optimization problem, with as specificity that the danger-level for the UMS of turning over is effectively estimated in function of the potential trajectories and taken into consideration for the choice of the optimal movement strategy. As a result, optimal safe trajectories for all the agents of the fleet can be planned.

We validate the proposed approach in simulation in an application scenario [4] tied to the surveillance of the Belgian off-shore windmill parks. The Belgian territorial waters are a very densely populated maritime area, with reserved spaces for all actors, as presented in Figure 1, and it is important that all actors stay within the delimited zones. For the windmill farms (area shaded in red on Figure 1), this often presents a problem, as other users (fishermen, pleasure yacht sailors, ...) penetrate this zone without permission. There is thus a need to patrol this area of around 10km x 30km.

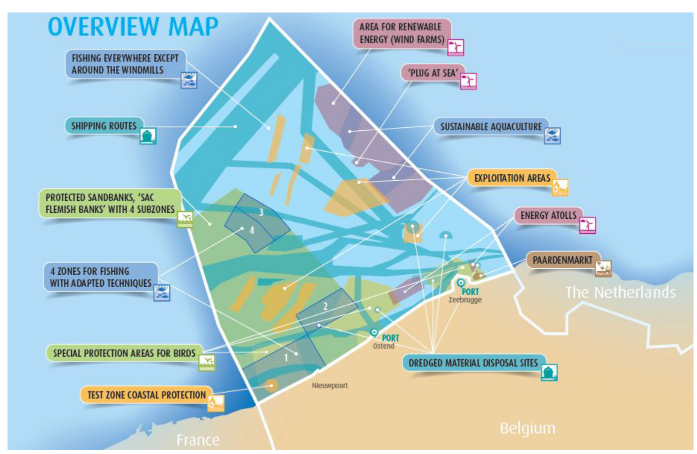

Fig. 1. Maritime Spatial Plan of the Belgian territorial waters (Source: Belgian Federal public service Health)

II. Previous Work

Multi-agent robotic coverage optimization is a research topic which has received a lot of attention in recent years, as more and more robotic assets are being deployed and thus also the need for coordination among these agents increases.

A first distinction to be made between the different methodologies is based upon the type of agents that is taken into consideration. On one hand, there are approaches that tackle *swarms* of a high number of less intelligent agents [5]. Swarm approaches generally make use of some form of ant colony optimization algorithm [6] for solving the coverage problem. On the other hand, there are *multi-agent* approaches that deal with a lower amount of more intelligent agents, which is the case in our application.

A second important distinction between methodologies is based upon the assumption which is made related to the connectivity between the different agents. If continuous broadband access between the agents is assumed, then all agents can get perfect localization and sensor data from one another and then the approaches are often based on some kind of global optimization approach [7], with the capability to adapt to a time-dependent environment [8]. Even though it has been shown that finding a globally optimal solution for the coverage maximization of a multi-agent fleet is an NP-hard problem [9], it is possible to come quite close to this solution within real-time constraints [10], [11].

If, on the other hand, unreliable network connections are assumed, then the agents cannot rely on a global planner and a local optimization is required. This also entails that a distributed approach is required which still allows for timely coordination between the different agents, as proposed by Xin et al. in [12].

Our methodology adopts a hybrid approach. Conceptually, it is based on a global optimization, but which is executed separately by each of the agents, taking into consideration the latest known data from the other agents. We use spatio-temporal memories to track and predict the localization and sensor data from the other agents, in order to cover up communication delays and breakdowns. Obviously, these estimations are not perfect, but in this way the optimization scheme tries to adopt the best of both kind of approaches.

Within the robotics community, most attention has been focused on providing solutions to the multi-agent coverage optimization problem for unmanned ground vehicles, but there are certainly also approaches that consider unmanned aerial vehicles [13]. However, for maritime systems, the research domain is less developed. Fabbri et al presented in [14] a path and decision support system for maritime surveillance vessels, based on multi-objective optimization algorithms that see to find an optimal trade-off among several mission objectives. While the concepts are similar, this paper focuses on a high-level decision support system for large manned vessels. In our application, we are interested in developing a solution for small-scale unmanned patrol vessels, which means that the requirements and constraints are very different.

III. Methodology

A. Overall framework

The proposed methodology methodology draws inspiration from behaviour-based control frameworks [15], where multiple behaviours actively work together to control the robot, or in this case the UMS. The main problem in behaviour-based control is how to synergize the different individual behaviours into a consistent and optimal global behaviour of the robotic agent. Therefore, we propose in this paper to use an optimization scheme to find the optimal weights, taking into consideration two objectives: increasing the global coverage (and thereby increasing the acquisition of new knowledge about the environment), and also minimizing the danger level (and thereby minimizing the chance for the vessel to capsize).

A major design issue for the development of such an optimization scheme is that the weight parameters to be optimized are subject to a large amount of environmental factors, such as the visibility, the wave height, etc. Therefore, we adopted a dual approach.

- **In an offline learning stage**, depicted by Algorithm 1, we repeatedly run an optimization process in order to find the optimal weight parameters w_{opt} for multiple environmental conditions:

$$w_{opt} = \arg\min_{w} \phi(w, \alpha, x, y, \theta, \nu, \gamma, v_{max}, \theta_{max}, w_h, w_\theta, om, \lambda) \tag{1}$$

  with:
  - w the weight parameters to be optimized
  - α the number of agents
  - (x, y) the position of the agents in a metric grid
  - $\theta[rad]$ the orientation of the agents
  - $\nu[m]$ the visibility, which is a function of the sensorial visibility (which is considered to be static, as the sensor package of the UMS does not change throughout a mission) and the meteorological visibility (which is dynamic, as the weather conditions may change throughout a mission).
  - $\gamma[rad]$ the field of view of the sensors on board of the UMS. The sensors are always assumed to be front-facing.
  - $v_{max}[m/s]$ the maximum velocity that can be obtained by the UMS
  - $w_h[m]$ the wave height
  - $x_\theta[rad]$ the wave orientation
  - om an obstacle map which is expressed as a probability density function
  - λ a parameter regulating the relative importance of coverage maximisation and danger minimisation

  The parameters of the optimization function ϕ are further explained in section III.C. For this optimization process, we used a quite classic Nelder-Mead simplex algorithm [16]. This process typically takes a very long time (a few days, depending on the granularity / resolution requested) and the resulting data is stored in a database for later retrieval.

- **In an online stage**, we retrieve the correct weight parameters for the environmental conditions at hand from the database and apply these directly to the same optimization function used before, as depicted by Algorithm 2.

In the following section, we will discuss in detail both parts of the optimization scheme.

B. Off-line optimization

Algorithm 1 depicts the off-line optimization scheme. As explained, its objective is to fill a database containing for each possible combination of environmental factors the optimal weight parameters.

Here, we focus on 4 main factors that have (experimentally) shown to have an important impact on the choice of the weight parameters: the Number of Assets (α), the visibility (ν), the wave height (w_h) and the wave direction (w_θ).

Concerning the Number of Assets (α), we consider fleets of 2 to 10 unmanned systems. The reason why this does not scale up further is that the methodology relies on an analysis of the localization and sensor data from all other assets. The methodology aims to predict the outcome of moving in a number of directions for each of these assets with is an $\mathcal{O}(N^2)$ problem. As a result, increasing the number of assets above 10 would lead to prohibitively long computation times.

Concerning the visibility, due to the fact that we consider the use of small (and thus low) vessels, the maximum visibility range is set to be 1000 meters.

In terms of wave height, the database considers wave heights up to 10 meters, even though the simulations show that the danger level for such big wave heights is very high and thus the seaworthiness is not really assured.

Algorithm 1 Off-line optimization

for $\alpha \leftarrow 2$ to 10 **do**
  for $nu \leftarrow 100$ to 1000 **do**
    for $w_h \leftarrow 0$ to 10 **do**
      for $w_\theta \leftarrow 0$ to 2π **do**
        $w \leftarrow \textsc{Optimization}(\phi)$
        $WeightsDatabase \leftarrow w$
      end for
    end for
  end for
end for

Algorithm 2 On-line optimization

1: $w \leftarrow WeightsDatabase$
2: $\textsc{InitializeAssets}()$
3: **for** $Iterations \leftarrow 1$ to $MaxIterations$ **do**
4:   **for all** $Assets$ **do**
5:     $[x_c, y_c] \leftarrow \textsc{SelectCandidatePos}(x_0, y_0, v_{max}, \theta)$
6:     **for all** $[x, y] \in [x_c, y_c]$ **do**
7:       $p_1 \leftarrow \textsc{EvaluateNewInfo}(x, y, x_0, y_0, cm, \nu, \gamma)$
8:       $p_2 \leftarrow \textsc{EvaluateBoatSpeed}(x, y, x_0, y_0)$
9:       $p_3 \leftarrow \textsc{EvaluateBoatOrientation}(x, y, x_0, y_0, \theta)$
10:       $p_4 \leftarrow \textsc{EvaluateWaves}(x, y, x_0, y_0, w_h, w_\theta)$
11:       $p_5 \leftarrow \textsc{EvaluateObstacles}(x, y, om)$
12:       $p_6 \leftarrow \textsc{SwarmOptimization}(x, y, x_i, y_i)$
13:       $p \leftarrow \textsc{Fuse}(w, p1, p2, p3, p4, p5, p6)$
14:       $p \leftarrow \textsc{ConstrainToBoundaries}(p, x, y, v, x_l, y_l)$
15:       $p \leftarrow \textsc{RemoveVisited}(p, x, y, Trajectory)$
16:       $[x_b, y_b] \leftarrow \textsc{ChooseMaximal}(p)$
17:     **end for**
18:     $[x, y, v, \theta] \leftarrow \textsc{MoveUMS}(x_b, y_b, x_0, y_0)$
19:     $danger \leftarrow \textsc{EstimateDanger}(\phi_w, v, w_h, w_\theta, \theta)$
20:     $cm \leftarrow \textsc{Sense}(x, y, \nu, \theta, \gamma)$
21:   **end for**
22:   $f_c \leftarrow \overline{cm}$
23:   $f_d \leftarrow \sum danger/\alpha$
24: **end for**
25: $f \leftarrow 1/f_c + \lambda f_d$

C. On-line optimization

Algorithm 2 depicts the on-line localisation scheme, which coincides with the optimization function ϕ of Algorithm 1. Each step of the pseudo-code algorithm is here explained:

1) In a first step, the relevant weights are extracted from the database. In case no exact match can be found, an interpolation is performed taking into consideration the closest matching conditions in the database.
2) The assets perform an initial communication to get to know each other's position. An empty coverage map (cm) is constructed. Note that we assume no a priori knowledge whatsoever. The robotic assets collectively build up a world model (a coverage map indicating areas they have visited and an obstacle map showing areas where they have found obstacles), that is maintained in memory by each of them (in order to be able to cope with network outages). This world model is empty at the start. The only information the assets have at the start is the position of each other and the boundaries of the working area.
3) Main loop for the simulation timer
4) Interrogate all UMS in the fleet
5) Choose a set of candidate positions (x_c, y_c) where the UMS can possibly move to. This depends on the starting position (x_0, y_0) and orientation θ and on the maximum velocity v_{max} of the UMS.
6) Explore all possible candidate positions
7) Assess the new information that can be retrieved by moving from the starting position (x_0, y_0) to the new position (x, y). This is achieved by adopting a visibility model, indicating, in function of the visibility ν and the sensor field of view γ, the probability of detecting an object in function of the vessel orientation. Figure 2 shows as an example a visibility model for a vessel that is oriented at a 45° angle at $(0, 0)$. This visibility model is compared to the coverage map (cm), which results in a local map p_1, which can be regarded as a heat map indicating what locations would be best to move to

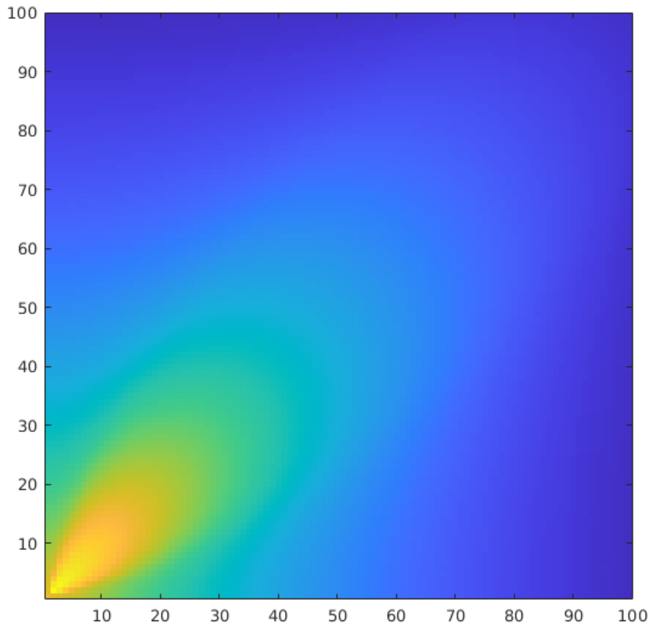

Fig. 2. Visibility model for a vessel that is oriented at a 45° angle at $(0, 0)$.

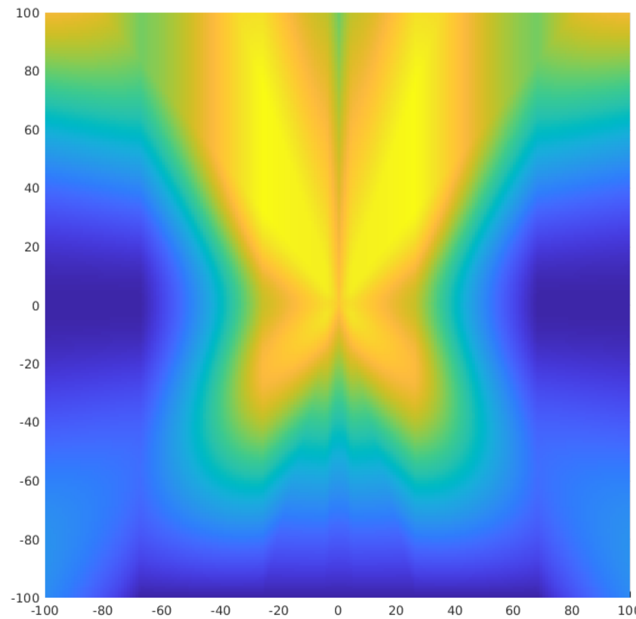

Fig. 3. Wave function for a vessel at location $(0, 0)$ and with incoming waves from the north side.

in order to obtain a maximum amount of new data (or otherwise said, to maximally increase the total value of the coverage map).

8) In order to maximize the chances of finding threats, it's better to move fast. However, the vessel should also not move too fast as this would not be fuel-efficient and could lead to incidents. Therefore, another function generates a local heat map favouring a good compromise vessel velocity v.
9) Vessels cannot change their orientation θ instantly. Therefore, another 'behaviour' generates a local heat map, avoiding these sharp turns.
10) Small vessels are extremely susceptible to waves. Both the wave height (w_h) and the wave direction (w_θ) play an important role and these need to be carefully aligned with the vessel speed and orientation. In order to assess this, we compiled on the basis of sailor knowledge in literature an empirical "wave-function" that expresses the danger level related to sea waves. This wave function is expressed as:

$$\phi_{wave} = (1 - y) * v * w_h \tag{2}$$

with y defined as:

$$y = 0.35x^6 - 3.5x^5 + 12.74x^4 - 20.75x^3 + 14.36x^2 - 2.9x + 0.1;$$

and with x defined as $x = \|\theta - w_\theta\|$.
Figure 3 shows an example of a wave function equation for a vessel at location $(0, 0)$ and with incoming waves from the north side. As you can notice, the most ideal orientantion for the vessel would be slightly inclined, but near head on to the waves. Orientations that are to be avoided are waves coming from the side or from the back.
11) Vessels should not run into detected obstacles. Therefore, the UMS create an obstacle map om and steer away from items on this map.
12) It is of no use that multiple agents of the fleet investigate the same area. Therefore the swarm optimization behaviour seeks to keep adequate distances between all of the agents.
13) The different local heat maps are combined into a single map p using the weights as calculated before.
14) An extra check is performed in order to ensure that the UMS do not stray away from the designated surveillance area x_l, y_l.
15) An extra check is made in order to avoid revisiting recent locations. Therefore, a trajectory memory is maintained and checked for pruning the local heat map p.
16) On the local heat map p, the optimal position x_b, y_b is located.
17) All possible positions are now checked.
18) The vessel is steered towards the optimal position.
19) The danger level for moving to this new position is estimated, based upon the wave function. The danger level is here defined as: $danger = 1 - \phi_{wave}$
20) The UMS performs an update of its sensing cycle, which will lead to an update of the coverage map, as new information is obtained.
21) End of the iteration over all agents.
22) The mean coverage score f_c is recorded
23) The total (summed) danger score f_d is recorded. For reasons of normalisation, it is divided by the number of

assets α.

24) End of the temporal loop.

25) We need to maximize the coverage, while minimizing the danger level. Therefore, the objective function to be minimized is defined as $f = 1/f_c + \lambda f_d$. The first term ensures that the coverage is maximized, while the second term ensures that the danger level is minimized. The parameter λ regulates the relative importance accorded to both aspects. This parameter is dependent on the type of vessel used. For smaller UMS, sea waves present a much higher risk, so λ should be higher. For larger vessels, λ can be reduced in order to maximize the coverage mapping quicker.

IV. Validation

For the validation of the proposed approach, we chose the application of the surveillance of the Belgian off-shore windmill parks, which means that an area of around 10km x 30km needs to be patrolled. However, the proposed methodology would for example also be very useful for a maritime search and rescue scenario [17] or a fishery control scenario.

In order to validate the methodology, we compared it to 5 state of the art solutions:

- **Random search**, where each agent adopts a completely random movement pattern
- **Distributed random search**, where the search area is subdivided in equal parts and each agent adopts a random search pattern within the designated subzone
- **Lawnmower search**, where each agent uses a movement pattern typically adopted by robotic lawnmowers: moving in straight lines and turning a random amount of degrees when coming near the boundaries
- **Distributed lawnmower search**, where the search area is subdivided in equal parts and each agent adopts a lawnmower search pattern within the designated subzone
- **Distributed Greek patterns**. This is the search and surveillance approach typically adopted by manned vessels and it has been proven to be very efficient for rapid area coverage. Moreover, by subdividing the search area and distributing the search tasks among multiple agents, this approach is quite well suited for maritime coverage optimization.

One disadvantage of all these state of the art approaches is that they do not take into consideration the danger posed by the waves on the vessel, which is an integral part of the proposed solution.

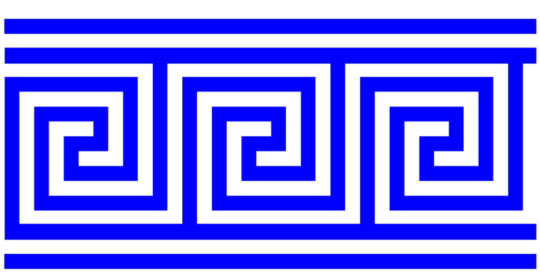

Fig. 4. Example of the Greek pattern

In order to further validate the optimization scheme, we compared also the results from a non-optimized, **nominal** version (static initial guess for the weights parameter w) with the optimized approach.

Figure 5 presents the results in terms of coverage in a simulation with 4 agents present. It can be clearly noted that the presented approach (denoted as **optimal** and indicated in dark red) achieves the highest overall coverage. Without using weight optimization, the Distributed Greek Patterns approach outperforms our baseline nominal approach here. All other approaches achieve a performance which is far lower.

These results can be expected, as the random search and lawnmower search approaches are quite simplistic methodologies, whereas the Greek Patterns has a proven track record for these kinds of applications. Still, using weight optimization, our proposed methodology succeeds in achieving a higher coverage score.

However, the major strength of our approach can be witnessed by also considering Figure 6, which indicates the danger level of executing a mission using each of the approaches. The blue portion of the bar chart indicates the mean danger level, whereas the red portion indicates the maximum danger level attained during a particular mission. Obviously, both are important to assess the risk of incidents. It can be clearly noted that both the nominal and the optimal proposed methodology achieve a danger level that is significantly lower than the other approaches. Moreover, for the optimal approach, there is little difference between the mean and the maximum danger levels, indicating that the methodology succeeds in keeping the risk at a constant and low level.

V. Conclusions

In this paper, we have presented an approach towards distributed coverage optimization for a maritime surveillance application. The approach is based upon a mix of off-line learning and on-line optimization. The methodology was validated by comparing it in simulation to multiple state of the art approaches. A next step will be to implement and test the system on real-life Unmanned Maritime Systems.

Acknowledgment

The research presented in this paper has been funded by the Belgian Royal Higher Institute for Defense, in the framework of the DAP19/08 (MarSur) project and by the Flemish Agency for Innovation and Entrepreneurship, in the framework of the Blue Cluster project SSAVE (HBC.2019.0045).

References

[1] L. Snidaro, I. Visentini, and K. Bryan, "Fusing uncertain knowledge and evidence for maritime situational awareness via markov logic networks," *Information Fusion*, vol. 21, pp. 159 – 172, 2015.

[2] E. Schwarz, D. Krause, H. Daedelow, and S. Voinov, "Near real time applications for maritime situational awareness," in *Deutscher Luft- und Raumfahrtkongress 2015*, pp. 1–5, November 2015.

[3] R. Csernatoni, "Constructing the eu's high-tech borders: Frontex and dual-use drones for border management," *European Security*, vol. 27, no. 2, pp. 175–200, 2018.

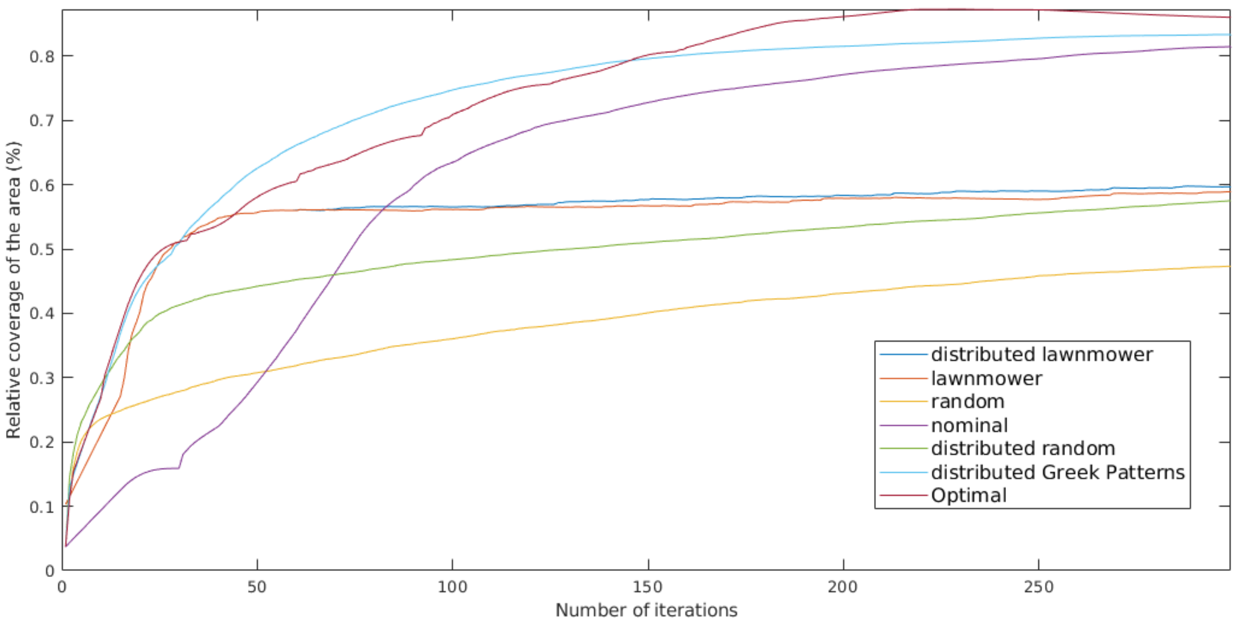

Fig. 5. Evolution of the relative coverage of a surveillance area using 7 different approaches.

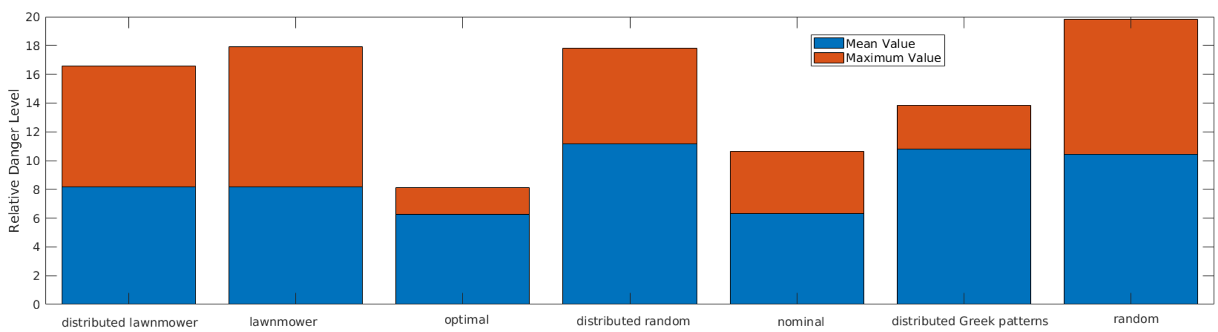

Fig. 6. Relative danger level for executing a maritime surveillance mission using 7 different approaches.

[4] G. De Cubber and R. Haelterman, "Optimized distributed scheduling for a fleet of heterogeneous unmanned maritime systems," in *International Symposium for Measurement and Control in Robotics (ISMCR)*, 2019.

[5] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, pp. 1–41, Mar 2013.

[6] M. Dorigo and T. Stutzle, *Ant Colony Optimization: Overview and Recent Advances*, pp. 311–351. Cham: Springer Int. Publishing, 2019.

[7] A. Pierson, L. C. Figueiredo, L. C. Pimenta, and M. Schwager, "Adapting to sensing and actuation variations in multi-robot coverage," *Int. J. of Robotics Research*, vol. 36, no. 3, pp. 337–354, 2017.

[8] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, and G. A. S. Pereira, *Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks*, pp. 85–99. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[9] C. Gao, Y. Kou, Z. Li, A. Xu, Y. Li, and Y. Chang, "Optimal multirobot coverage path planning: Ideal-shaped spanning tree," *Mathematical Problems in Engineering*, vol. 2018, p. 3436429, Sep 2018.

[10] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, "Darp: Divide areas algorithm for optimal multi-robot coverage path planning," *J. of Intelligent & Robotic Systems*, vol. 86, pp. 663–680, Jun 2017.

[11] J. Cortes, "Coverage optimization and spatial load balancing by robotic sensor networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 3, pp. 749–754, 2010.

[12] B. Xin, G. Gao, Y. Ding, Y. Zhu, and H. Fang, "Distributed multi-robot motion planning for cooperative multi-area coverage," in *Int. Conf. on Control Automation (ICCA)*, pp. 361–366, 2017.

[13] J. Valente, J. Del Cerro, A. Barrientos, and D. Sanz, "Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach," *Computers and Electronics in Agriculture*, vol. 99, pp. 153 – 159, 2013.

[14] T. Fabbri, R. Vicen-Bueno, R. Grasso, G. Pallotta, L. M. Millefiori, and L. Cazzanti, "Optimization of surveillance vessel network planning in maritime command and control systems by fusing metoc ais vessel traffic information," in *OCEANS 2015 - Genova*, pp. 1–7, 2015.

[15] D. Doroftei, G. De Cubber, E. Colon, and Y. Baudoin, "Behavior based control for an outdoor crisis management robot," in *Int. Ws. on Robotics for Risky Interventions and Environmental Surveillance*, 2009.

[16] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the nelder-mead simplex method in low dimensions.," *SIAM J. Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[17] G. De Cubber, D. Doroftei, K. Rudin, K. Berns, D. Serrano, J. Sanchez, S. Govindaraj, J. Bedkowski, and R. Roda, "Search and rescue robotics-from theory to practice," 2017.