



Distributed coverage optimisation for a fleet of unmanned maritime systems

Geert De Cubber¹, Rihab Lahouli², Daniela Doroftei¹, Rob Haelterman²

¹ Royal Military Academy, Department of Mechanics, Avenue de la Renaissance 30, 1000 Brussels, Belgium

² Royal Military Academy, Department of Mathematics, Avenue de la Renaissance 30, 1000 Brussels, Belgium

ABSTRACT

Unmanned maritime systems (UMS) can provide important benefits for maritime law enforcement agencies for tasks such as area surveillance and patrolling, especially when they are able to work together as one coordinated system. In this context, this paper proposes a methodology that optimises the coverage of a fleet of UMS, thereby maximising the opportunities for identifying threats. Unlike traditional approaches to maritime coverage optimisation, which are also used, for example, in search and rescue operations when searching for victims at sea, this approach takes into consideration the limited seaworthiness of small UMS, compared with traditional large ships, by incorporating the danger level into the design of the optimiser.

Section: RESEARCH PAPER

Keywords: Unmanned maritime systems; multi-agent systems; maritime surveillance; distributed coverage optimisation

Citation: Geert De Cubber, Rihab Lahouli, Daniela Doroftei, Rob Haelterman, Distributed coverage optimisation for a fleet of unmanned maritime systems, Acta IMEKO, vol. 10, no. 3, article 7, September 2021, identifier: IMEKO-ACTA-10 (2021)-03-07

Section Editor: Bálint Kiss, Budapest University of Technology and Economics, Hungary

Received January 15, 2021; **In final form** September 9, 2021; **Published** September 2021

Copyright: This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Funding: The research presented in this paper has been funded by the Belgian Royal Higher Institute of Defence, 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).

Corresponding author: Geert De Cubber, e-mail: geert.de.cubber@rma.ac.be

1. INTRODUCTION

An ever-increasing percentage of the global population lives in coastal areas. A downside of this evolution is that an increasing number of criminals are turning their attention to our seas and oceans to carry out illegal activities. Examples include drug smuggling, human trafficking, illegal fishery and border infringements. The problem for law enforcement agencies is that patrolling and surveilling the vast oceans using traditional means (large, manned vessels) is impossible from an economic and operational point of view.

Unmanned maritime systems (UMS) can potentially provide maritime law enforcement agencies with a valuable tool for increasing their capabilities in relation to maritime surveillance. Of course, UMS are not the only answer; they are just one part of a much wider maritime situational awareness toolkit [1], which also encompasses satellite monitoring [2], manned and unmanned aerial assets [3] with advanced analytics solutions, allowing the data gathered by all these agents to be turned into information and knowledge.

One of the main capabilities the UMS require is to be able to operate as a well-coordinated group, 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 operational requirements of the end users [4], the interoperability between the different systems [5] and towards the design of the surveillance architecture. As an example, the traditional approaches towards distributed patrol and surveillance [6]-[8] by manned systems generally 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, a novel methodology for the real-time control of a fleet of between two and ten UMS will therefore be proposed. The presented methodology is cast as a distributed coverage optimisation problem that specifies that the danger level for the UMS of overturning is effectively estimated as a function of the potential trajectories and considered in the selection of the optimal movement strategy. As a result, the optimal safe trajectories for all the agents in the fleet can be planned.



Figure 1. Maritime spatial plan of Belgian territorial waters, showing the very dense occupation of these waters by different actors and for different economic activities. This paper considers the surveillance and patrol of offshore wind farms, indicated on the map as the area with a red overlay (Source: Belgian Federal Public Service – Health, Food Chain Safety and Environment).

The proposed approach is validated through a simulation in an application scenario [9] connected to the surveillance of Belgian offshore wind farms. 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 wind farms (area shaded in red on Figure 1), this often presents problems, as other users (e.g. fishing vessels and pleasure yachts) penetrate this zone without permission. In order to police and enforce the exclusion zone, it is necessary to patrol this area, which is on the maritime border with The Netherlands and measures about 10 km by 30 km.

2. PREVIOUS STUDIES

Multi-agent robotic coverage optimisation is a research topic that has received a considerable amount of attention in recent years, as an increasing number of robotic assets are being deployed; thus, the need to identify strategies to optimise the coordination between these agents has increased.

A first distinction to be made between the different methodologies is based upon the type of agents that are taken into consideration. On the one hand, there are approaches that tackle *swarms* of a high number of less intelligent agents [10]. Swarm approaches generally make use of some form of ant colony optimisation algorithm [11] to solve the coverage problem. On the other hand, there are *multi-agent* approaches that deal with a lower number of more intelligent agents, which is the case for the application in the present study.

A second important distinction between methodologies is based upon the assumptions made with regard to the connectivity between the different agents. If continuous broadband access between the agents is assumed, then all agents can obtain perfect localisation and sensor data from one another, and then the approaches can be based on some kind of global optimisation approach [12], with the capability to adapt to a time-dependent environment [13]. Even though it has been shown that finding a globally optimal solution for the coverage maximisation of a multi-agent fleet is an NP-hard problem [14], it is possible to come quite close to this solution within real-time constraints [15], [16]; however, this requires intelligent strategies to guide the optimisation process (see more discussion on this subject later).

If, however, unreliable network connections are assumed, then the agents cannot rely on a global planner, and a local optimisation is required. This also entails the need for a distributed approach that still allows for timely coordination between the different agents within the system, as proposed by Xin et al. [17].

The methodology presented here adopts a hybrid approach. Conceptually, it is based on a global optimisation, but one that is executed separately by each of the agents, taking into consideration the latest known data from the other agents. Spatio-temporal memories are used to track and predict the localisation and sensor data from the other agents in order to address communication delays and breakdowns. Clearly, these estimations are not perfect, but in this way, the optimisation scheme tries to adopt the best of both types of approach.

Within the robotics community, most attention has been focused on providing solutions to the multi-agent coverage optimisation problem for unmanned ground vehicles, but there are certainly also approaches that consider unmanned aerial vehicles [18]. However, for maritime systems, the research domain is less developed. Fabbri et al. [19] presented a path and decision support system for maritime surveillance vessels, based on multi-objective optimisation algorithms that seek to find an optimal trade-off between several mission objectives. While the concepts are similar, this paper focuses on a high-level decision support system for large, manned vessels. This application aims to develop a solution for small-scale, unmanned patrol vessels, which means that the requirements and constraints are very different.

As discussed previously, finding a globally optimal solution for the coverage maximisation of a multi-agent fleet is an NP-hard problem [14]. This implies that the algorithms scale up traditionally badly for an increasing number of agents. As the hybrid planner proposed here features a mix of global and local optimisation aspects, it also suffers from the drawback of global planning systems. In order to remedy this problem, researchers have suggested particle optimisation [20], as proposed by Han et al., or grey wolf optimisation methodologies, as first introduced by Mirjalili et al. [21] and later improved for distributed coverage optimisation problems by Wang et al. [22]. In short, all these methodologies aim to intelligently prune the number of candidate positions that have to be investigated in order to limit the number of computations to be performed. Developing further on these ideas, an optimisation strategy is also proposed that quickly selects the high-probability candidate positions, thereby limiting the computation time.

3. METHODOLOGY

3.1. Overall framework

The proposed methodology draws inspiration from behaviour-based control frameworks [23], in which 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 synergise the different individual behaviours into a consistent and optimal global behaviour for the robotic agent. This requires the choice of so-called weight parameters that lead to the expected global behaviour. However, finding these weight parameters is a non-trivial task. Therefore, this study proposes an optimisation scheme to find the optimal weights, taking into consideration two objectives: a) increasing the global coverage (and thereby increasing the acquisition of new knowledge about the environment) and b) minimising the danger level (thereby minimising the likelihood of the vessel capsizing).

A major design issue for the development of such an optimisation scheme is that the weight parameters to be optimised are subject to a large number of environmental factors, such as visibility and wave height. Therefore, for the present study, a dual approach was adopted.

- **At the offline learning stage**, depicted by algorithm 1, an optimisation process was repeatedly run to find the optimal weight parameters w_{opt} for multiple environmental conditions:

$$w_{opt} = \arg \min_w \phi(w, \alpha, x, y, \theta, v, \gamma, v_m, \theta_m, w_h, w_\theta, om, \lambda), \quad (1)$$

with the following parameters:

- w represents the weight parameters to be optimised.
- α is the number of agents.
- (x, y) is the position of the agents in a metric grid.
- θ is the orientation of the agents in radians.
- v is the visibility in meters. This is a function of the sensorial visibility (which is considered to be static, as the UMS sensor package does not change during a mission) and the meteorological visibility, which is dynamic, as the weather conditions may change throughout a mission.
- γ is the sensors' field of view on board the UMS (rad). In this implementation, the sensors are always assumed to be front facing (although the field of view can be set to 360 °).
- v_m is the maximum velocity (m/s) that can be reached by the UMS.
- θ_m is the maximum turning rate (rad/s) that can be achieved by the different UMS.
- w_h is the wave height (m).
- w_θ is the wave orientation (rad).
- om is an obstacle map, expressed as a probability density function, that expresses the probability of finding an obstacle.
- λ is a dimensionless parameter regulating the relative importance of coverage maximisation and the minimisation of the risk of capsizing.

The parameters of the optimisation function ϕ and the function itself are further explained in Section 3.3. For this optimisation process, the classic Nelder–Mead

simplex algorithm [24] was used. This process typically takes a long time (a few days, depending on the granularity or resolution requested). For this reason, Section 3.4 introduces an accelerated optimisation scheme. At the end of this process, the resulting data were stored in a database for later retrieval (during the online stage).

- **At the online stage**, the correct weight parameters for the environmental conditions at hand were retrieved from the database and applied directly to the same optimisation function used before, as depicted by algorithm 2.

In the following section, both parts of the optimisation scheme will be discussed in detail.

3.2. Offline optimisation

Algorithm 1 depicts the offline optimisation scheme. As explained, its objective is to develop a database with the optimal weight parameters for each possible combination of environmental factors. This study has focused on four main factors that have been experimentally shown to have an important impact on the choice of the different weight parameters: the number of assets α , visibility v , wave height w_h and wave direction w_θ .

With regard to the number of assets α , fleets of between two and ten unmanned systems have been considered for this study. The reason why this number cannot be scaled up further is that the methodology relies on an analysis of the localisation 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 an $\mathcal{O}(N^2)$ problem. As a result, increasing the number of assets above ten leads to prohibitively long computation times, at least for the non-optimised version of the algorithm (see Section 3.4 for details).

Concerning visibility, as this study is based on the use of small vessels (which have a minimal height and therefore a limited view over the waves), the maximum visibility range is set to 1,000 m.

In terms of wave height, the database considers wave heights up to 10 m even though the simulations show that the danger level for such large wave heights is very high, and thus, the seaworthiness of the UMS considered in this implementation is not assured.

3.3. Online optimisation

Algorithm 2 depicts the online localisation scheme, which coincides with the optimisation function ϕ of algorithm 1.

Each step of the pseudo-code algorithm is explained here in detail:

Algorithm 1 Offline optimisation

```

1: for  $\alpha \leftarrow 2$  to 10 do
2:   for  $nu \leftarrow 100$  to 1000 do
3:     for  $w_h \leftarrow 0$  to 10 do
4:       for  $w_\theta \leftarrow 0$  to  $2\pi$  do
5:          $w \leftarrow \text{OPTIMISATION}(\phi)$ 
6:          $\text{WeightsDatabase} \leftarrow w$ 
7:       end for
8:     end for
9:   end for
10: end for

```

Algorithm 1. Offline optimisation scheme.

1. First, the relevant weights are extracted from the database. If no exact match can be found, an interpolation is performed taking into consideration the closest matching conditions in the database developed during the offline stage.
2. The assets perform an initial communication to get to know each other's position. An empty coverage map (cm) is constructed (note that no a priori knowledge is assumed). 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), which is maintained in memory by each of them in order to be able to cope with network outages. This world model is initially empty; the only information the assets have at the start is each other's position and the boundaries of the working area.
3. The main loop for the simulation timer is created.
4. All the UMS in the fleet are interrogated.
5. A set of candidate positions (x_c, y_c) that the UMS are able to move to is selected, depending on the starting position (x_0, y_0), orientation θ and the maximum velocity $\bar{v}_{max}(v_m, \theta_m)$ of the UMS.
6. All possible candidate positions are explored.
7. New information that can be retrieved by moving from the starting position (x_0, y_0) to the new position (x, y) is assessed. This is achieved by adopting a visibility model, indicating, through visibility v and the sensor field of view γ , the probability of detecting an object as a result of the vessel's orientation. The adopted visibility model assumes a mix of infrared, visual and LIDAR-based sensing and draws upon the heuristically established sensor models established by Lahouli et al. [25] (for infrared sensors) and Balta et al. [26] (for visual and

LIDAR sensors). Figure 2 provides an example of a visibility model for a vessel that is oriented at a 45° angle at the position $(0,0)$. This visibility model is compared to the coverage map (cm), resulting in a local map p_1 , which can be regarded as a heat map indicating the best locations to move to in order to obtain the maximum amount of new data (i.e. to maximally increase the total value of the coverage map).

8. In order to maximise the chances of finding threats, it is better to move fast. However, the vessel should not move too fast because this would not be fuel efficient, and it could lead to incidents. Therefore, another function generates a local heat map favouring a compromise vessel velocity v .
9. Vessels are not able to change their orientation θ suddenly. Therefore, another 'behaviour' generates a local heat map that avoids 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, an empirical 'wave function' was compiled, based on sailors' experiences set out in the literature, that expresses the danger level related to waves. This wave function is expressed as

$$\phi_{wave} = (1 - y) v w_h, \quad (2)$$

with y defined as

$$y = 0.35 x^6 - 3.5 x^5 + 12.74 x^4 - 20.75 x^3 + 14.36 x^2 - 2.9 x + 0.1. \quad (3)$$

Figure 3 provides an example of a wave-function equation for a vessel at location $(0,0)$ and with incoming waves from the north. As can be seen, the ideal orientation for the vessel (highest value of ϕ_{wave}) would be slightly inclined but nearly head on to the waves. Orientations that are to be avoided (lowest value of ϕ_{wave}) are waves coming from the side or from the back.

11. It is important that vessels do not run into any detected obstacles. The UMS therefore collectively create and share an obstacle map (om) and steer away from objects on this map.
12. It would be inefficient for multiple agents in the fleet to investigate the same area. Therefore, the swarm optimisation behaviour seeks to maintain an adequate distance between the agents.
13. The different local heat maps are combined into a single map p using the weights that were calculated previously (in the offline step).
14. An extra check is performed in order to ensure that the UMS do not stray away from the designated surveillance area.
15. An extra check is made in order to avoid revisiting recent locations. This is required not only to speed up the convergence but also to avoid getting stuck at local minima. 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 then checked.
18. The vessel is steered towards the optimal position.

Algorithm 2 Online optimisation

```

1:  $w \leftarrow WeightsDatabase$ 
2: INITIALISEASSETS()
3: for  $Iterations \leftarrow 1$  to  $MaxIterations$  do
4:   for all  $Assets$  do
5:      $[x_c, y_c] \leftarrow SELECTCANDIDATEPOS(x_0, y_0, v_{max}, \theta)$ 
6:     for all  $[x, y] \in [x_c, y_c]$  do
7:        $p_1 \leftarrow EVALUATENEWINFO(x, y, x_0, y_0, cm, v, \gamma)$ 
8:        $p_2 \leftarrow EVALUATEBOATSPEED(x, y, x_0, y_0)$ 
9:        $p_3 \leftarrow EVALUATEBOATORIENTATION(x, y, x_0, y_0, \theta)$ 
10:       $p_4 \leftarrow EVALUATEWAVES(x, y, x_0, y_0, w_h, w_\theta)$ 
11:       $p_5 \leftarrow EVALUATEOBSTACLES(x, y, om)$ 
12:       $p_6 \leftarrow SWARMOPTIMISATION(x, y, x_i, y_i)$ 
13:       $p \leftarrow FUSE(w, p_1, p_2, p_3, p_4, p_5, p_6)$ 
14:       $p \leftarrow CONSTRAINTTOBOUNDARIES(p, x, y, v, x_l, y_l)$ 
15:       $p \leftarrow REMOVEVISITED(p, x, y, Trajectory)$ 
16:       $[x_b, y_b] \leftarrow CHOOSEMAXIMAL(p)$ 
17:     end for
18:      $[x, y, v, \theta] \leftarrow MOVEUMS(x_b, y_b, x_0, y_0)$ 
19:      $danger \leftarrow ESTIMATEDANGER(\phi_w, v, w_h, w_\theta, \theta)$ 
20:      $cm \leftarrow SENSE(x, y, v, \theta, \gamma)$ 
21:   end for
22:    $f_c \leftarrow \overline{cm}$ 
23:    $f_d \leftarrow \sum danger/\alpha$ 
24: end for
25:  $f \leftarrow 1/f_c + \lambda f_d$ 

```

Algorithm 2. Online optimisation scheme.

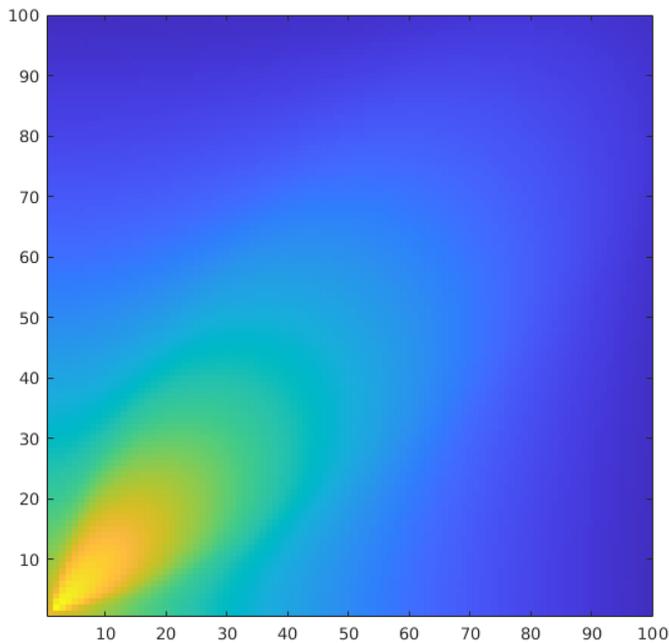


Figure 2. Visibility model for a vessel that is oriented at a 45 ° angle at the position (0,0).

19. The danger level for moving to this new position is estimated based on 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 update the coverage map as new information is obtained.
21. The iteration for all agents ends.
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. The temporal loop ends.
25. Coverage needs to be maximised while minimising the danger level. Therefore, the objective function to be minimised is defined as

$$f = \frac{1}{f_c} + \lambda f_d. \quad (4)$$

The first term of the objective function ensures that the coverage is maximised, while the second term ensures that the danger level is minimised. 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 maximise the coverage mapping more rapidly.

3.4. Computational speed optimisation

As can be seen in the definition of algorithm 2, the computation time rises exponentially with the number of unmanned systems considered. Not only do all of these assets require a separate evaluation (step 4 in algorithm 2), but also the complexity of many of the subprocesses (e.g. swarm optimisation) increases with the number of agents. The main culprits for the long computation time after this are the number of candidate positions that have to be evaluated (step 5 in

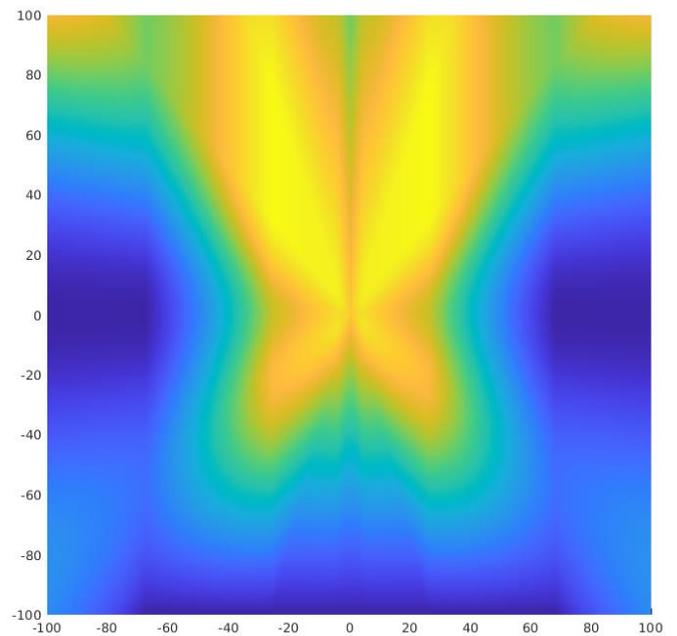


Figure 3. Wave function for a vessel at location (0,0) and with incoming waves from the north.

algorithm 2) and the sometimes-high number of iterations required for the convergence of the Nelder–Mead simplex algorithm, used for solving (1).

The adopted optimisation methodology is geared primarily towards an intelligent pruning of the candidate positions, as follows:

- In the first phase, only a very limited subset of the original candidate positions is selected. This is performed by downscaling the selected candidate positions on a lower-scale metric grid.
- In the second phase, an analysis (as described in algorithm 2, lines 6 to 17) is performed on the downscaled candidate positions.
- A local subgrid is defined at the original resolution around the resulting position.
- For all candidate positions within this local subgrid, the analysis (as described in algorithm 2, lines 6 to 17) is performed again and a final position is selected.

The double loop may seem to add extra complexity and computation time, but in practice, it avoids the evaluation of a large number of candidate positions that do not have a viable chance of being selected. Indeed, as the local maps are mostly continuous functions, it makes sense to evaluate them first at a lower resolution and then to scale up. Furthermore, the convergence settings of the Nelder–Mead simplex algorithm were optimised to match the candidate position pruning approach.

4. VALIDATION

4.1. Quantitative validation

For the validation of the proposed approach, an application was selected that is used for the surveillance of Belgian offshore wind farms, which have an area of around 10 km × 30 km that needs to be patrolled. However, the proposed methodology would also be very useful for a maritime search and rescue [27] or a fishery control scenario.

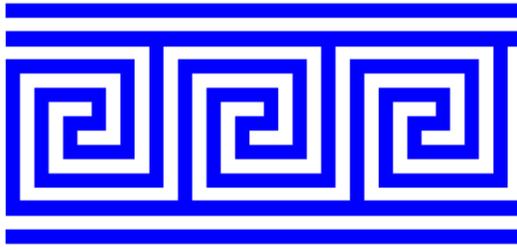


Figure 4. Example of distributed Greek patterns.

In order to validate the methodology, it was compared to five state-of-the-art solutions:

- **A random search**, where each agent adopts a completely random movement pattern,
- **A distributed random search**, where the search area is subdivided into equal parts and each agent adopts a random search pattern within the designated subzone,
- **A lawnmower search**, where each agent uses a movement pattern typically adopted by robotic lawnmowers: moving in straight lines and turning a random number of degrees when approaching boundaries,
- **A distributed lawnmower search**, where the search area is subdivided into equal parts and each agent adopts a lawnmower search pattern within the designated subzone, and
- **Distributed Greek patterns**. This is the search and surveillance approach typically adopted by manned vessels, which 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 to maritime coverage optimisation. Figure 4 provides an example of the Greek pattern.

One disadvantage of all these state-of-the-art approaches is that they do not take into consideration the danger the waves

pose to the vessel, which is an integral part of the proposed solution.

In order to further validate the optimisation scheme, the results were also compared using a non-optimised, **nominal** version (using a static initial estimate for the weight parameter w) with the optimised approach.

Figure 5 presents the results in terms of coverage in a simulation with four agents present. It can be clearly seen that the presented approach (denoted as **optimal** and indicated in dark red) achieves the highest overall coverage. Without using weight optimisation, the distributed Greek patterns approach outperforms the baseline nominal approach presented here. All other approaches achieve a performance that is far lower. It can also be observed that the coverage results do not always monotonically increase. This can be explained by the fact that at each iteration, the existing coverage data is 'aged' (in practice, the coverage map is multiplied by 0.99) in order to represent the fact that older data have become less valid. The result is that with a limited number of agents, it becomes very difficult to maintain a high overall coverage score.

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

However, the major strength of this approach can be seen in 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. Clearly, both are important for assessing the risk of incidents. It can be clearly seen that both the nominal and the optimal proposed methodology achieve a danger level that is significantly lower than in 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 maintaining risk at a constant and low level.

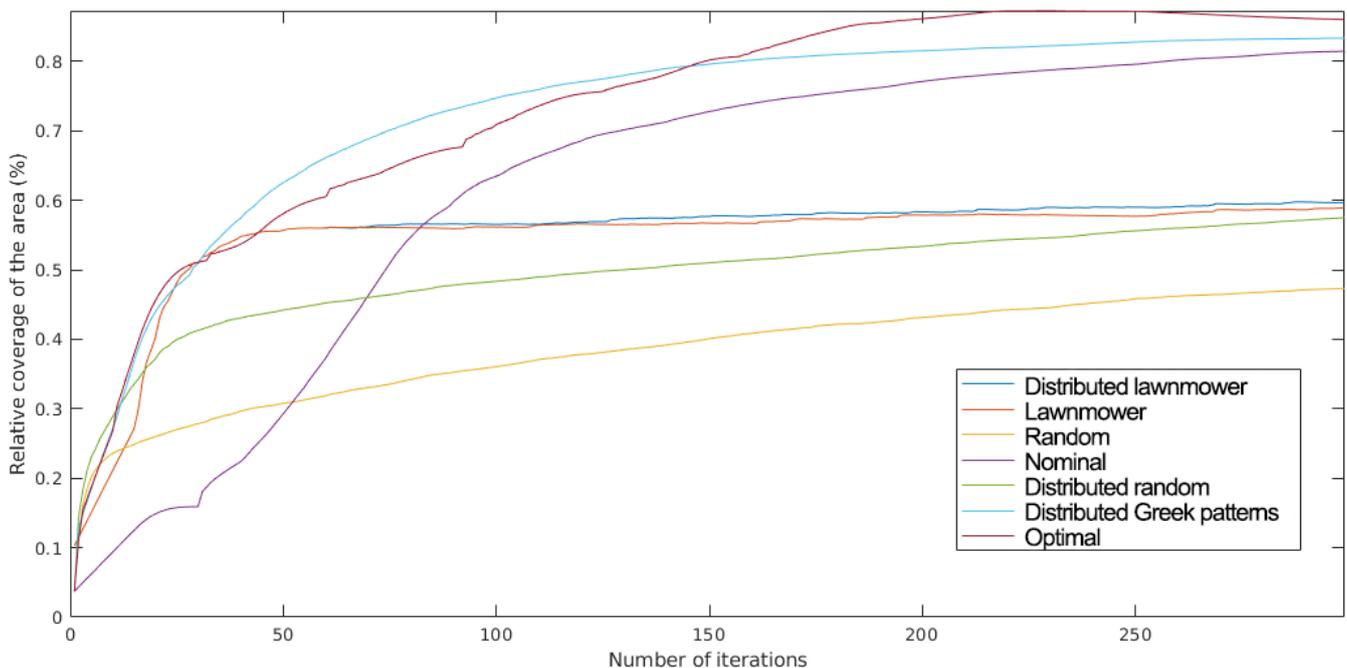


Figure 5. Evolution of the relative coverage of a surveillance area using seven different approaches.

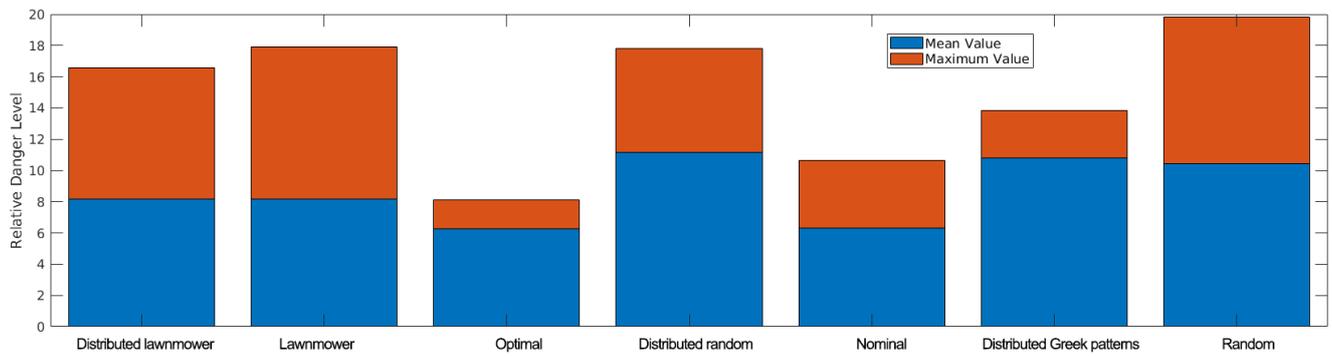


Figure 6. Relative danger level for executing a maritime surveillance mission using seven different approaches.

4.2. Scaling and timing

In order to assess the effects of the speed optimisation methodology explained in Section 3.4, Figure 7 shows the evolution of the processing time, relative coverage and relative danger level for the proposed approach with and without the application of the candidate location pruning methodology.

As is clearly indicated in Figure 7, the computation time is drastically reduced by the incorporation of the candidate position pruning methodology. In general, the accelerated approach is about 9 times faster than the baseline approach. This clearly shows that the proposed methodology of first analysing a limited set of points and then providing a detailed analysis of a dense point set in just one small area has a highly beneficial impact on the global processing time. This also enables the methodology to be used for an increased number of assets even though the global algorithm still scales up to slightly more than $\mathcal{O}(N)$.

An important aspect to assess is whether there would be any loss of quality using the accelerated approach. In order to evaluate this, the coverage mapping and danger levels were recorded for both methodologies and for the different numbers of agents, as shown in Figure 7.

Surprisingly, the accelerated approach performed even better than the baseline approach even though the differences were not large. This may seem counter-intuitive at first sight because the accelerated approach evaluates fewer candidate positions and thus, compared with the baseline approach, always has a higher risk of being trapped at local minima.

This phenomenon was investigated and found to be caused by the better convergence properties of the accelerated approach. Indeed, as the accelerated approach requires fewer time-consuming local map evaluation steps, the Nelder–Mead simplex optimisation algorithm achieves (slightly) lower values for the optimisation function ϕ when using the accelerated method. The ‘side effect’ of accelerated optimisation, as described in Section 3.4, is therefore also an improved (higher) coverage mapping and a slightly improved (lower) danger level.

5. CONCLUSIONS

In this paper, an approach towards distributed coverage optimisation for a maritime surveillance application has been presented. The approach is based upon a mix of offline learning and online optimisation. In order to remedy the traditional problem related to the excessive processing time for multi-agent global planning methodologies, an approach for the multi-scale selection of candidate locations has also been proposed.

The methodology was validated by comparing it in simulation to multiple state-of-the-art approaches. This comparison demonstrated that the proposed approach performed well in

terms of coverage mapping and very well in terms of minimising the danger of capsizing for small, unmanned vessels. Moreover, the validation of the performance of the accelerated approach on multi-agent systems demonstrated that the computation time can be drastically reduced while the coverage mapping performance is increased.

A next step will be to implement and test the system on the real-life UMS that are planned to be deployed to patrol Belgian territorial waters.

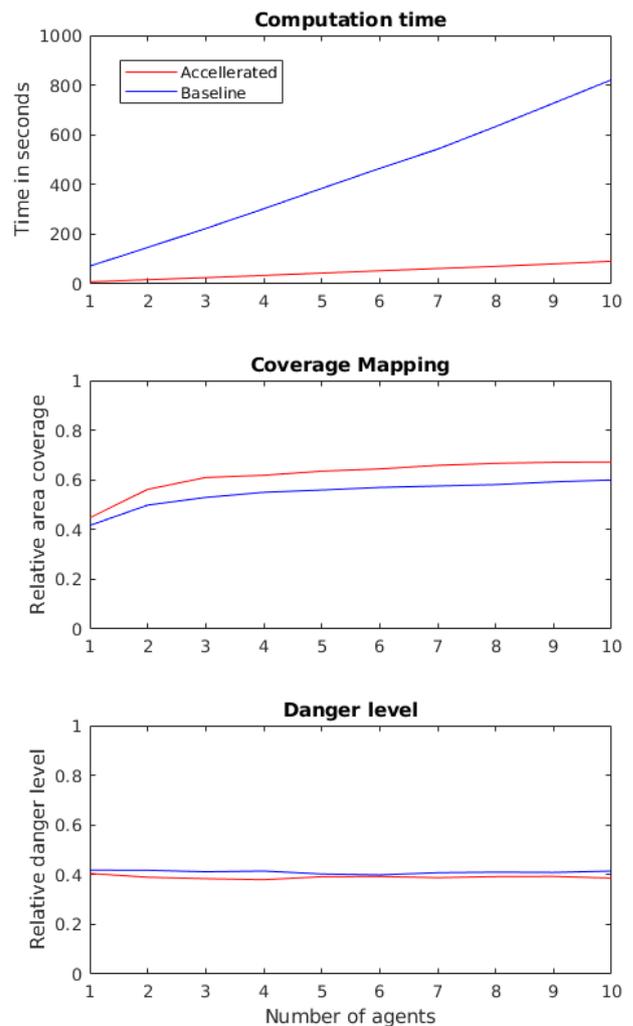


Figure 7. Evolution of the processing time, mean relative coverage and mean relative danger level for the proposed approach with and without the application of the candidate location pruning methodology.

ACKNOWLEDGEMENT

The research presented in this paper has been funded by the Belgian Royal Higher Institute of Defence, 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, K. Bryan, Fusing uncertain knowledge and evidence for maritime situational awareness via Markov logic networks, *Information Fusion* 21 (2015), pp. 159-172. DOI: [10.5281/zenodo.22949](https://doi.org/10.5281/zenodo.22949)
- [2] E. Schwarz, D. Krause, H. Daelow, S. Voinov, Near real time applications for maritime situational awareness, 36th International Symposium on Remote Sensing of Environment, Berlin, Germany, 11–15 May 2015, pp. 1-5. DOI: [10.5194/isprsarchives-XL-7-W3-999-2015](https://doi.org/10.5194/isprsarchives-XL-7-W3-999-2015)
- [3] R. Csernatoni, Constructing the EU's high-tech borders: Frontex and dual-use drones for border management, *European Security* 27 (2018), pp. 175-200. DOI: [10.1080/09662839.2018.1481396](https://doi.org/10.1080/09662839.2018.1481396)
- [4] D. Doroftei, A. Matos, G. De Cubber, Designing search and rescue robots towards realistic user requirements, *Applied Mechanics and Materials*, vol. 658 (2014), pp. 612-617. DOI: [10.4028/www.scientific.net/AMM.658.612](https://doi.org/10.4028/www.scientific.net/AMM.658.612)
- [5] D. S. López, G. Moreno, J. Cordero, J. Sanchez, S. Govindaraj, M. M. Marques, V. Lobo, S. Fioravanti, A. Grati, K. Rudin, M. Tosa, A. Matos, A. Dias, A. Martins, J. Bedkowski, H. Balta, G. De Cubber, Interoperability in a heterogeneous team of search and rescue robots, in: *Search and Rescue Robotics - From Theory to Practice*. G. De Cubber, D. Doroftei (editors), InTech, Rijeka, 2017, ISBN 978-953-51-3375-9, pp. 93-125. DOI: [10.5772/intechopen.69493](https://doi.org/10.5772/intechopen.69493)
- [6] C. Yan, T. Zhang, Multi-robot patrol: a distributed algorithm based on expected idleness, *Int. J. of Advanced Robotic Systems* 13 (2016), pp. 1-12. DOI: [10.1177/1729881416663666](https://doi.org/10.1177/1729881416663666)
- [7] D. Aksaray, K. Leahy, C. Belta, Distributed multi-agent persistent surveillance under temporal logic constraints, *IFAC-PapersOnLine* 48 (2015), pp. 174-179. DOI: [10.1016/j.ifacol.2015.10.326](https://doi.org/10.1016/j.ifacol.2015.10.326)
- [8] M. Basoggio, A. Cenedese, P. Merlo, M. Pozzi, L. Schenato, Distributed perimeter patrolling and tracking for camera networks, *Proc. of the 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, 15-17 December 2010, pp. 2093-2098. DOI: [10.1109/CDC.2010.5717883](https://doi.org/10.1109/CDC.2010.5717883)
- [9] G. De Cubber, R. Haelterman, Optimized distributed scheduling for a fleet of heterogeneous unmanned maritime systems, *Proc. of the International Symposium for Measurement and Control in Robotics (ISMCR)*, 19-21 September 2019, pp. A3-2-1-A3-2-7. DOI: [10.1109/ISMCR47492.2019.8955727](https://doi.org/10.1109/ISMCR47492.2019.8955727)
- [10] M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: a review from the swarm engineering perspective, *Swarm Intelligence* 7 (2013), pp. 1-41. DOI: [10.1007/s11721-012-0075-2](https://doi.org/10.1007/s11721-012-0075-2)
- [11] M. Dorigo, T. Stutzle, Ant colony optimization: overview and recent advances. In: M. Gendreau, J. Y. Potvin (eds.). *Handbook of Metaheuristics*. International Series in Operations Research & Management Science, vol 146. Springer, Boston, MA., USA, 2019, DOI: [10.1007/978-1-4419-1665-5_8](https://doi.org/10.1007/978-1-4419-1665-5_8)
- [12] A. Pierson, L. C. Figueiredo, L. C. Pimenta, M. Schwager, Adapting to sensing and actuation variations in multi-robot coverage, *Int. J. of Robotics Research* 36 (2017), pp. 337-354. DOI: [10.1177/0278364916688103](https://doi.org/10.1177/0278364916688103)
- [13] L. C. A. Pimenta, M. Schwager, Q. Lindsey, V. Kumar, D. Rus, R. C. Mesquita, G. A. S. Pereira, Simultaneous Coverage and Tracking (SCAT) of Moving Targets with Robot Networks, Springer, Berlin Heidelberg, 2010, ISBN 978-3-642-00311-0.
- [14] C. Gao, Y. Kou, Z. Li, A. Xu, Y. Li, Y. Chang, Optimal multirobot coverage path planning: ideal-shaped spanning tree, *Mathematical Problems in Engineering* vol. 2018, 2018, p. 3436429. DOI: [10.1155/2018/3436429](https://doi.org/10.1155/2018/3436429)
- [15] A. C. Kapoutsis, S. A. Chatzichristofis, E. B. Kosmatopoulos, Darp: divide areas algorithm for optimal multi-robot coverage path planning, *J. of Intelligent & Robotic Systems* 86 (2017), pp. 663-680. DOI: [10.1007/s10846-016-0461-x](https://doi.org/10.1007/s10846-016-0461-x)
- [16] J. Cortes, Coverage optimization and spatial load balancing by robotic sensor networks, *IEEE Transactions on Automatic Control* 55 (2010), pp. 749-754. DOI: [10.1109/TAC.2010.2040495](https://doi.org/10.1109/TAC.2010.2040495)
- [17] B. Xin, G. Gao, Y. Ding, Y. Zhu, H. Fang, Distributed multi-robot motion planning for cooperative multi-area coverage, *Proc. of the Int. Conf. on Control Automation (ICCA)*, Ohrid, Macedonia, 3-6 July 2017, pp. 361-366. DOI: [10.1109/ICCA.2017.8003087](https://doi.org/10.1109/ICCA.2017.8003087)
- [18] J. Valente, J. Del Cerro, A. Barrientos, D. Sanz, Aerial coverage optimization in precision agriculture management: a musical harmony inspired approach, *Computers and Electronics in Agriculture* 99 (2013), pp. 153-159. DOI: [10.1016/j.compag.2013.09.008](https://doi.org/10.1016/j.compag.2013.09.008)
- [19] T. Fabbri, R. Vicen-Bueno, R. Grasso, G. Pallotta, L. M. Millefiori, L. Cazzanti, Optimization of surveillance vessel network planning in maritime command and control systems by fusing METOC & AIS vessel traffic information, *Proc. of OCEANS 2015 - Genova, Italy*, 18 – 21 May 2015, pp. 1-7.
- [20] X. Han, S. Li, X. Pang, Coverage optimization algorithm of wireless sensor network, in: *Advances in Future Computer and Control Systems. Advances in Intelligent and Soft Computing*, vol. 159. D. Jin, S. Lin (editors). Springer, Berlin, Heidelberg, 2012, ISBN 9783642293870, pp. 32-40.
- [21] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Advances in Engineering Software* 69 (2014), pp. 46-61. DOI: [10.1016/j.advengsoft.2013.12.007](https://doi.org/10.1016/j.advengsoft.2013.12.007)
- [22] Z. Wang, H. Xie, Z. Hu, D. Li, J. Wang, W. Liang, Node coverage optimization algorithm for wireless sensor networks based on improved grey wolf optimizer, *J. of Algorithms & Computational Technology* 13 (2019), pp. 1-15. DOI: [10.1177/1748302619889498](https://doi.org/10.1177/1748302619889498)
- [23] D. Doroftei, G. De Cubber, E. Colon, Y. Baudoin, Behavior based control for an outdoor crisis management robot, *Proc. of the Int. Ws. on Robotics for Risky Interventions and Environmental Surveillance*, Brussels, Belgium, 12-14 January 2009, 8 pp.
- [24] J. C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright, Convergence properties of the Nelder–Mead simplex method in low dimensions, *SIAM J. Optimization* 9 (1998), pp. 112-147. DOI: [10.1137/S1052623496303470](https://doi.org/10.1137/S1052623496303470)
- [25] I. Lahouli, E. Karakasis, R. Haelterman, Z. Chtourou, G. De Cubber, A. Gasteratos, R. Attia, Hot spot method for pedestrian detection using saliency maps, discrete Chebyshev moments and support vector machine, *IET Image Processing* 12 (2018), pp. 1284-1291. DOI: [10.1049/iet-ipr.2017.0221](https://doi.org/10.1049/iet-ipr.2017.0221)
- [26] H. Balta, J. Velagic, G. De Cubber, W. Bosschaerts, B. Siciliano, Fast statistical outlier removal based method for large 3D point clouds of outdoor environments, *Proc. of the 12th IFAC Symposium on Robot Control - SYROCO 2018*, Budapest, Hungary, 27-30 August 2018, vol. 51, issue 22, pp. 348-353. DOI: [10.1016/j.ifacol.2018.11.566](https://doi.org/10.1016/j.ifacol.2018.11.566)
- [27] G. De Cubber, D. Doroftei, K. Rudin, K. Berns, D. Serrano, J. Sanchez, S. Govindaraj, J. Bedkowski, R. Roda, Search and Rescue Robotics - from Theory to Practice, InTech, Rijeka, 2017, ISBN 978-953-51-3375-9.