Resource Allocation in Time-Constrained Environments: The Case of Frigate Positioning in Anti-Air Warfare

J-F. Morissette[1], B. Chaib-draa[2], P. Plamondon[3]

Computer Science & Software Engineering Department
Laval University
Ste-Foy, PQ, Canada.

[1]*jfmoris@damas.ift.ulaval.ca,* [2]*chaib@damas.ift.ulaval.ca,*
[3]*plamon@damas.ift.ulaval.ca*

ABSTRACT *Maritime environments are known to be very complex environments with tight real-time constraints where it is very difficult to manage resource allocation. This is the case, for example, for a frigate which must position itself in order to use its resources the most effectively possible to increase its chances of survival when the time of air raids comes. Under such very hard constraints, it can often happen that the commander makes errors because of the complexity of the environment or the stress which the situation can generate. We propose here to implement a decision-aid system which suggests the position that the frigate must take. We start by giving an heuristic which evaluates the effectiveness of a position according to the threats found in the environment. Then, we propose an algorithm which treats all the possible rotations and suggests the best regarding a given situation. Finally, we expose the results of our experiments and we comment on them.*

KEYWORDS Command and Control, Decision-Aid, Frigate Movement, Resource Management

1 Introduction

Maritime environments are known to be very complex environments with tight real-time constraints. In case of frigate attack, the commander must make fast decisions by considering several factors to ensure himself of the best possible survival of the frigate and its crew. Under such real-time constraints, it can often happen that the commander makes errors because of the complexity of the environment or the stress which the situation can generate. In these conditions, a computer is tremendously faster than a human and consequently, it can suggest decisions in time thus facilitating the task of a commander.

In this paper, we propose a decision-aid system which suggests to the commander the best position for the frigate in case of attack. Starting from threats which the frigate faces, the system proposes a position which will make it possible for the frigate to use its resources the most effectively possible in order to increase its chances of survival. We start by making a description of the decision-aid system. Next, we give an heuristic which makes it possible to evaluate the effectiveness of the possible positions. Then, we propose a technique which allows reducing the number of possible rotations of the frigate in order to decrease complexity. Finally, we show the results of our experiments.

2 Description of the Decision-Aid System

As we explained in the previous section, the system that we propose here is a system for decision support. Its observations are the threats which are impending on the frigate. For our simulation, we assume that everything that appears in the operating range of the frigate radar is seen. Starting from the position of the threats, the system must choose the position which makes it possible for the frigate to defend itself in the best possible way by making an effective use of its resources. To achieve that, it proposes a rotation which will bring the threats into the most effective sectors of defense. The difficulty comes from the fact that the agent must give a response within a very short time. Thus, time available for the agent to make its decision is very short. The algorithm, for the choice of the position, proposed in this paper is an algorithm in the order of n^2 (noted $O(n^2)$). So, with such an algorithm, the answer will be given very quickly by the agent.

3 The Potential Number of Hits Heuristic (PNH)

The model we used for the frigate resources is a considerably simplified model of the relevant AAW hardkill and softkill weapons that we can find on real frigates. Details of the model are described in [1]. To understand this paper, we only have to know that some resources have restrictions and they cannot be used all around the frigate.

3.1 Defense Sectors

Because some weapons have a blind zone, we can divide the frigate defense area into 12 sectors. Because some sectors have exactly the same resources, it is possible to regroup them in order to have fewer sectors to consider by the learning module. Figure 1 presents the 12 defense sectors and the graphic of results given by the learning module explained in the next subsection.

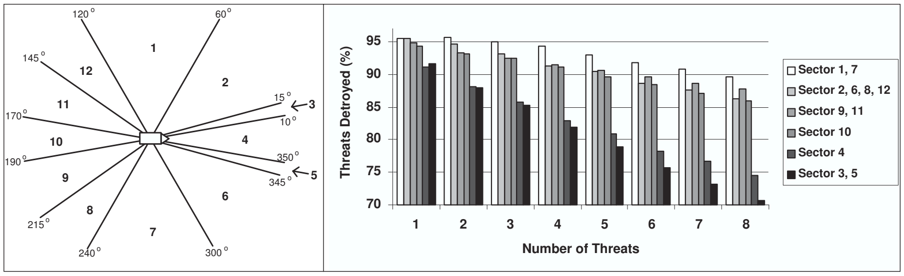

Figure 1. Division into Sectors and Efficiency of Each Sector.

3.2 Learning Module

The learning module is used to calculate the defense effectiveness of each sector according to the number of threats found in that sector. The effectiveness of each sector is the percentage of threats destroyed in this sector according to the number of threats in it. Construction of the learning module was carried out as follows. For each sector, we did 8000 simulations per number of threats. Simulations were made for 1 to 8 threats, all threats appearing in the sector to be tested. So, for each sector, we did a total of 64000 simulations. Based on these results, we calculated for each sector the percentage of threats that were destroyed according to the number of threats being found there.

3.3 The PNH Heuristic

The heuristic exposed here is used to evaluate the effectiveness of a frigate position by considering threats which attack the frigate. It is an heuristic which calculates the potential number of threats which will hit the frigate by taking into account the number of threats being found in each sector. We call this heuristic the Potential Number of Hits (PNH) heuristic. Calculation is carried out as follows:

$$PNH = \sum_{i=1}^{12} n_i \cdot (1 - H(i|n_i)) \quad (1)$$

where n_i represents the number of threats found in the sector i and $(1-H(i|n_i))$ represents the percentage of threats that reached the frigate when n_i threats are in the sector i. The different values of $H(i|n_i)$ are given by the learning module. Because $H(i|n_i)$ represents the percentage of threats destroyed, we should calculate $(1 - H(i|n_i))$ to obtain the percentage of threats that reached the frigate.

4 The Reduction of Complexity

In this section, we explain the model and the algorithm to treat all the possible cases. Let us start with the following definition:

A *threats-sectors combination* is a set of n couples formed in this way:

$$(1, s_1), (2, s_2), (3, s_3), ..., (n, s_n) \qquad \text{where} \quad 1 \leq s_i \leq 12 \quad \text{for} \quad 1 \leq i \leq n$$

The first coordinate represents the ID of the threat and the second coordinate represents the ID of the sector where it is.

Before giving the method, we simply give an idea of how it works, this will help to understand. Starting from the initial state, we want to generate all the possible threats-sectors combinations. In this way, we should consider all the rotations included in $[-180, 180]$ elaborating thus the list of all the possible threats-sectors combinations. However, considering all rotations in $[-180, 180]$ would lead us to a very large computational complexity. Our method exposes a way of finding all the threats-sectors combinations without however falling into a large computational complexity. When all the threats-sectors combinations are found, we evaluate them according to the PNH heuristic. When the best combination is found, we calculate the minimum and maximum rotations the frigate can do to obtain this combination.

4.1 Threats-Sectors Combinations Generator (GenTSC)

First of all, we must specify that the time available for the rotation is known. Moreover, we know the rotation speed of the frigate. Therefore, it is possible to know the maximum rotation which the frigate can carry out. Let R_{max} be the maximum rotation.

For each threat, we look in which sector the threat would be found by making a rotation from R_{max} and $-R_{max}$. We thus obtain, for each threat, the set S_i of sectors which can be reached within the time available for rotation. For each sector of S_i, we simulate a frigate rotation which would bring the threat to the beginning of this sector [1]. For example, in Figure 2, to bring the threat at the beginning of sector 2, we simulate a frigate rotation of 60°. Thus, for each threat, there is a maximum of 12 simulated rotations. Let n be the number of threats, we will thus have a maximum of $12n$ simulated rotations, where 12 represents the number of frigate defense sectors. The proof that this method simulates all the possible threats-sectors combinations is given below.

Thereafter, for each simulated rotation, we evaluate the threats-sectors combination generated by this rotation with the PNH heuristic explained in the

[1] The beginning of a sector is the angle we obtain when we enter the sector by making a counterclockwise rotation around the ship.

Section 3.3. Finally, we sort the threats-sectors combinations according to effectiveness. The calculation of real rotation that the frigate must carry out is explained in Section 4.2.

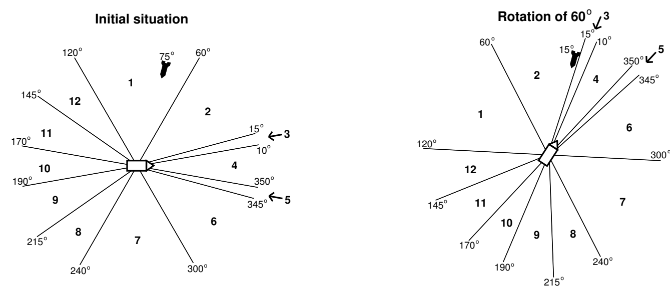

Figure 2. Rotation to Bring the Threat in Sector 2.

Proposition 1

The list of positions suggested by GenTSC will give us the set of all possible threats-sectors combinations according to the starting position of the threats and time available for rotation.

In other words, no matter the value of the rotation chosen in the interval $[-R_{max}, R_{max}]$, the threats-sectors combination obtained with this rotation can also be obtained with one of the rotations suggested by GenTSC.

Proof of Proposition 1

Let us take an unspecified threats-sectors combination which is able to be reached within the time available. Let us prove that this combination can be reached with one of the rotations suggested by GenTSC.

Let us define the following variables for each threat:

s_i: the sector of the threat i
θ_i: the angle of the beginning of the zone s_i
d_i: the angle of attack of the threat i relative to the frigate direction
$\Delta_i = d_i - \theta_i$

Now, let us define $\Delta_{min} = \underset{1 \leq i \leq n}{argmin}(\Delta_i)$

Let us consider $j = i$ such that $\Delta_i = \Delta_{min}$. Thus, j represents the ID of the threat having the smallest Δ_i.

By making a frigate rotation of Δ_j, the angle of the threat j becomes equal to the angle of the beginning of the zone s_j. So, after the Δ_j rotation, $d_j = \theta_j$.

Moreover, since $\Delta_j = \underset{1 \leq i \leq n}{argmin}(\Delta_i)$, we can affirm that with a rotation Δ_j, no threat will change of sector.

Lastly, because $d_j = \theta_j$ and the threats-sectors combination can be reached within the time available, this threats-sectors combination can be reached with a position suggested by GenTSC.

4.2 Method Final-Positioning (FinPos)

Now that the threats-sectors combinations are sorted according to PNH heuristic, we must determine which position the frigate must take. To achieve that, we must check that the best combination can be reached (a combination cannot be reached if the frigate does not have enough time to be able to reach this combination). If it cannot be reached, we check the second threats-sectors combination and so on until a combination can be reached.

The way of proceeding is as follows. By using the threats-sectors combination targeted, we know in which sector we must find each threat. Therefore, for each threat, we can calculate an interval of rotation to bring the threat in its concerned sector. This interval will be noted $[x_i, y_i]$ for the threat i. When all the intervals of rotation are calculated, we can calculate the interval of rotation of the frigate, noted $[R_{start},\ R_{end}]$, with the following formula:

$$[R_{start},\ R_{end}] = [\ \underset{1 \leq i \leq n}{argmax}(x_i),\ \ \underset{1 \leq j \leq n}{argmin}(y_j)\ ]$$

For the lower bound, we use *argmax* because it is the minimal rotation to bring each threat in its sector. Indeed, if we do not take *argmax* but rather a smaller rotation, at least one threat would not be in the right sector. A similar reasoning justifies the use of *argmin* for the upper bound.

Remark: As the targeted threats-sectors combination comes from a simulated rotation of the frigate, we can affirm that this combination can be reached and that $R_{start} \leq R_{end}$. But, this does not guarantee that this combination can be reached within the time available for the rotation.

Finally, we made a proof that the complete algorithm to find the best position is in the order of n^2, noted $O(n^2)$. This proof will be given in a longer version of the paper.

5 Experiments

In modern Anti-Air Warfare (AAW), proving a concept might be a difficulty in itself. Of course, given the price of weaponry and the sheer number of people involved, it is not practical to develop prototypes for new concepts on real ships. This is why we have conceived a sophisticated simulator called Naval Defense Simulator (NDS) which permits to test our new concepts.

5.1 Bench Tests

Tests were made with 3 types of movement: no movement, bayesian movement and PNH movement. When there is no movement, the frigate does not move at all. The bayesian and PNH movements use the algorithm explained in this paper. The difference is the heuristic used to calculate the efficiency of the suggested positions. The bayesian movement is explained in [2]. The PNH movement uses the PNH heuristic as explained in this paper.

Tests were also made with 3 types of planner: reactive, deliberative and deliberative with tabu improvement. Details on these planners are explained in [3]. Finally, tests were made for 1 to 8 threats. Since we have 3 types of movement, 3 types of planner and 1 to 8 threats, we thus have 72 different situations. For each situation, we simulate 5 000 scenarios. So, we made a total of 360 000 tests. Tests results are exposed in the left graphic of Figure 3. Results represent the percentage of threats that were destroyed. We should note that for each movement algorithm, results are the mean of the 3 planners.

Finally, we made tests to see the time taken by our algorithm on a computer. We tested the algorithm using the bayesian and PNH heuristics implemented in Java. Tests were made on an Intel Centrino 1.5 GHz with 512 Mb DDR333. For each number of threats, we simulate 200 different scenarios and each scenario called the algorithm 1000 times to be able to do a mean. Right graphic of Figure 3 shows these results in milliseconds.

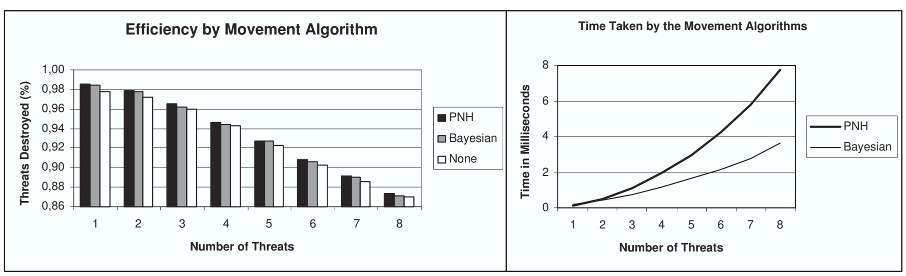

Figure 3. Efficiency of Algorithms and Time Taken to Give an Answer.

5.2 Discussion about results

Relative to the no movement, we can see that the bayesian and the PNH movements improve the defense a little bit. When the number of threats increases, the PNH movement gives better results because this heuristic takes into account the number of threats found in each sector. In PNH, the threats dependency is considered but this is not still optimal. For 1 to 8 threats, the means of the threats destroyed percentage are the following: 92.92% for no movement, 93.3% for the bayesian movement and 93.45% for the PNH movement. So, relative to the no movement, the bayesian movement increases the threats destroyed percentage of 5.37%. The PNH movement increases it of 7.49%.

Finally, as we can see in Figure 3, the time taken by both movement algorithms is under 10 milliseconds which is very good if we consider that we have between a few seconds to about two minutes to make the decision.

6 Conclusion and Future Work

In this paper, we exposed a method for positioning the frigate in case of threats attack. We saw that the method is able to treat every possible case. In particular, it permits reducing computation and we gave a proof that it does not reduce the quality of the solution. We also exposed an heuristic (PNH) which calculates the efficiency of a position. We saw that this heuristic takes into account all threats in the same sector. In this way, a part of the dependency between threats is considered. A possible improvement would be to find heuristics making it possible to consider not only the dependence between threats in the same sector but to consider the dependence with the threats in nearby sectors. This could be interesting because some sectors have the same resources and it can happen that a resource cannot be available because it is used on a threat in an other sector.

Also, having an interval of rotation rather than a fixed rotation allows more latitude to the commander. Indeed, if the commander knows that later he will have a new rotation to make, he could choose to make a rotation corresponding to the maximum bound of the interval which will permit saving time for the second rotation.

Finally, we could see that this decision-aid system can enormously help the commander of a frigate because of the very fast answer given by the system. An answer given in less than one second leaves sufficient time to the commander to analyze the solution suggested by the system and to adopt or not this solution thereafter.

REFERENCES

[1] Blodgett D., Plamondon P., Chaib-draa B., Kropf P., Bossé E.(2002), *A Method to Optimize Ship Maneuvers for the Coordination of Hardkill and Softkill Weapons within a Frigate,* 7^{th} International Command and Control Research and Technology Symposium (7^{th}ICCRTS)

[2] Plamondon, P. (2003), *A Frigate Survival Approach based on Real-time Multiagent Planning,* Master thesis, Computer Science & Software Engineering Department, Laval University

[3] Soucy, M. (2003), *Planification déliberative en temps réel,* Master thesis, Computer Science & Software Engineering Department, Laval University