Collaborative Combat: Multi-Radars Active Tracking Resources Allocation by Distributed Auctions

Pierre Larrenie
THALES SIX GTS, Gennevilliers
Univ. Gustave Eiffel, CNRS, Marne-la-Vallée
FRANCE

pierre.larrenie@esiee.fr

Cédric Buron
KLaIM,
L@bISEN, ISEN Ouest, Carquefou,
FRANCE

cedric.buron@isen-ouest.yncrea.fr

Frédéric Barbaresco
THALES LAND & AIR SYSTEMS
Meudon
FRANCE

frederic.barbaresco@thalesgroup.com

ABSTRACT

In this paper, we present an algorithm allocating tasks for a set of static autonomous radars with rotating antennas. It allows a set of radars to allocate in a fully decentralized way a set of tasks for tracking targets according to their location, considering that tracking a target with several can improve accuracy. The allocation algorithm proceeds through a collaborative and fully decentralized auction protocol, using a collaborative auction protocol (Consensus Based Bundle Auction algorithm). Our algorithm is based on a double use of our allocation protocol among the radars. The latter begin by allocating targets, then launch a second round of allocation if they have resources left, in order to improve accuracy on targets already tracked.

1.0 INTRODUCTION

Managing multiple sensors is a topic of rising interest. The coordination of multiple radars is very useful when the sensors are mobile, but even in the case of fixed radars, coordination can improve the overall performance of a radar system. This can be achieved in multiple ways: first, in cases where all the radars cannot track all the target in range, distributing the targets among the radars can allow to track more targets. But this is not the only advantage that can be taken from the coordination of radars. By choosing to allocate the same target to well chosen radars, it is also possible to improve the precision of the tracking by intersecting the information of the radars. Coordination can be achieved either in a centralized or in a decentralized way. Decentralized allocation allows to be more resilient, both in the case of a failure of the center and in the case of a communication issue between the center and the radars. In both cases, the fact that the computation is replicated and the fact that the radars communicate with multiple others allow the overall system to keep working despite potential faults. Performing such decentralized allocation falls in the field of multi-agent systems. In this paradigm, so called "agents" (here radars) act in an autonomous, reactive, proactive yet interactive way in order to achieve their goals. The social ability allows agents to communicate, hence to organize themselves in a decentralized way.

Multi-agents have been widely used in robot teams. One of the major issues regarding robots is the real-time constraint, which is also applicable in the context of radars. In this context, the main application methods are machine learning [2], market-based methods and DCOP [8]. In this paper we introduce a new method based on a double distributed action in order to allocate targets to radars in a tracking setting. Our approach allows to allocate the targets in an efficient way in real time, while allowing two radars to track the same target in order to improve the precision around the targets. In order to evaluate our approach, we compare it with a centralized approach, based on a centralized solver (Coin OR Branch and Cut, CBC) and show that the stability of the algorithm allows to perform better than the centralized approach.

The paper is organized as follows: section 2 introduces related works on decentralized task allocation for sensors, and in particular market-based approaches. We then formally introduce the collaborative multi-radar tracking problem in section 3. We present our market-based algorithm in section 4.

2.0 RELATED WORKS

Several works have focused on the use of decentralized approaches for task allocation for sensors, since the seminal work of Lesser et al. [3]. Since then, many approaches have been used, and recently, for real time use cases, auction methods have gained much interest in the multi-agent community [5], [8] for their capacity to perform good allocation in an affordable time. Many methods have been using auctions since then to allocate tasks in real-time, including to robots, sensors and radars (see for instance [10]).

One of the most successful recent algorithms is CBBA [11]. This algorithm allows to perform the allocation in a fully decentralized way, with agents acting both as auctioneers and bidders. This algorithm has since been used several times for sensors [1], [7]. However, none of them has taken into account the specificities of radars, *i.e.* their collaboration through the intersection of their uncertainty ellipses, or the challenge of high dynamicity.

3.0 THE COLLABORATIVE MULTI-RADAR TRACKING PROBLEM

3.1 Radar Model

In this section, we introduce the different elements of the problem. We first describe the model that has been used for the radars. We then introduce a definition of the multi-target multi-radar allocation problem in the formalism of Constraint Optimization Problems (COP).

We consider that each radar has a two-dimensional frame in a polar coordinate system centered on itself. The influence of elevation is negligible in our setting, so it is not useful to use a three-dimensional landmark. Each target therefore has a position in the radar reference frame determined by its distance, denoted r, and its azimuth (polar angle), denoted θ. The precision of the measurement made by the radar is noted: σ_r in distance and σ_θ in azimuth. The resulting measurement uncertainty is represented as an ellipse. The measure itself corresponds to a centered two-dimensional Gaussian random variable of covariance $K = R(\theta)\begin{pmatrix}\sigma_r & 0\\ 0 & \sigma_\theta\end{pmatrix}$ where $R(\theta)$ represents the rotation matrix of angle θ. During active tracking, the aim is to anticipate the next position of the target, based on its past positions and its current position, using a Kalman filter. There is also a prediction uncertainty associated with the measurements.

The radar assumes that the signal it receives is subject to Gaussian white noise. The signal-to-noise ratio (S/N)[1] is assumed to remain constant, with a value set to 13, which is a commonly used value in practice. The S/N has an impact on the standard deviation of the measurements and reflects the desired output quality specified by the user. With a given S/N, parameters such as the transmitted wavelength or transmission power can be selected.

3.2 Multi-Sensor Multi-Target Allocation Problem

The problem of optimal allocation of multiple radars and targets can be approached in various ways. If future actions are to be considered, we can model them through Dec-POMDP. However, this problem falls under the EXPTIME complexity class, and even heuristic methods may take a significant amount of time. Additionally, the targets being considered are highly mobile and unpredictable, making it challenging to anticipate future movements accurately. Therefore, we propose to formalize the problem of optimal allocation for each time step without taking future evolution into account. This problem can be formalized as a constraint optimization problem:

$$\max \sum_{i,j,k} c_{ijk} \cdot w_{ijk}$$

$$\text{such that: } \begin{cases} w_{ijk} = x_{M_{ij}} \wedge x_{O_{kj}}, \forall (i,k) \in \mathcal{I}^2, \forall j \in \mathcal{J} & (A_{ikj}) \\ \sum_i w_{ij} \leq 1, \forall j \in \mathcal{J} & (C2) \\ \sum_j \gamma_{ij} \cdot \left(X_{M_{ij}} + X_{O_{kj}} - w_{iij} \right) \leq L_{t_i} \forall i \in \mathcal{I} & (L) \\ \left(x_{M_{ij}}, x_{O_{kj}} \right) \in \{0,1\}^2,\ \forall (i,k) \in \mathcal{I}^2, \forall j \in \mathcal{J} & \\ w_{ikj} \in \{0,1\}, \forall (i,k) \in \mathcal{I}^2, \forall j \in \mathcal{J} & \end{cases}$$

where:

- $\mathcal{I}$ is the set of radars and $\mathcal{J}$ the set of tasks. Note that we are placed here, in the framework $\mathcal{I} \ll \mathcal{J}$.
- c_{ikj}: Corresponds to the utility that the radar i and the radar k provide to the system if the radar i handles the task j as a main radar and k as an optional radar. c_{ikj} is of the following form, with $V(E_{ij})$ (respectively $V(E_{kj})$) the surface of the ellipse E_{ij} (resp. E_{kj}) described by the matrix P_{ij} (resp. P_{kj}) of the Kalman filter of the radar i (resp. k) for the target j and $V(E_{ij} \cap E_{kj})$ the intersection volume of these two ellipses.
- $x_{M_{ij}}$: Boolean variable, $x_{M_{ij}}$ equals 1 if the radar i performs the task j as the main radar, 0 otherwise.
- $x_{O_{ij}}$: Boolean variable, $x_{O_{ij}}$ equals 1 if the radar i performs the task j as an optional radar, 0 otherwise.
- w_{ikj}: Boolean variable, w_{ikj} equals 1 if the radar i performs the task j as main radar and the radar k performs the task j as optional radar, 0 otherwise.

The constraints can be understood the following way:

- (A_{ikj}) defines w_{ikj} as i following the target j as main radar (we also write $x_{M_{ij}} = 1$), and k follows it as optional radar $x_{O_{kj}} = 1$. $w_{iik} = 1$ if there is only one radar i following the target. The operator $\wedge$ corresponds to the logical AND operator.

[1] The S/N (Signal to Noise Ratio) corresponds to the ratio between the power of the useful signal and the power of ambient noise.

- $(C2)$ lists all possible combinations of 2 sensors that track a target j. There is at most only one combination of sensors that can be chosen.
- (L) models the load of the radar. If the radar is tracking the target as main or optional radar, the term between parentheses equals 1, otherwise it equals 0 and the load for the task is therefore not considered.

There is a total of $|\mathcal{I}|^2 \cdot |\mathcal{J}| + |\mathcal{J}| + |\mathcal{I}|$ constraints. However, note that this formulation is challenging to generalize to a set of n sensors, as it would result in a significant increase in the number of constraints and Boolean variables. This, in turn, would make the problem unsolvable[2] for a classical solver. In this paper, we limit ourselves to two radars for each target. The representation of the uncertainty ellipses is provided on Figure 1. On this figure, the radar is represented as a dark blue circle, the target as a red triangle. The orange ellipse represents the uncertainty ellipse when the radar starts to track the target. The red ellipse represents the uncertainty ellipse, after several Kalman filter steps.

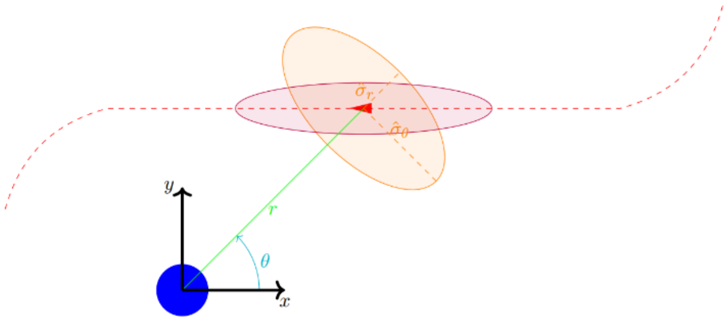

Figure 1: Illustration of uncertainty ellipses during active pursuit.

4.0 AUCTION-BASED MULTI-RADAR MULTI-TARGET ALLOCATION

The Consensus Based Bundle Auction algorithm (CBBA) [11] is an algorithm in which agents bid on a sequence of tasks based on the information they possess and share information with their neighbors. The algorithm can be divided into two phases that are repeated successively:

1. the bidding phase, during which agents propose a bid on a sequence (or bundle) of actions while attempting to optimize utility improvement based on the information they have regarding the current situation, and
2. the consensus phase, where agents share and receive information from neighbors and adjust their bundle based on the newly acquired information.

4.1 The CBBA Algorithm

The messages that agents send to each other can be represented as a set of vectors. The set of vectors that an agent sends to another corresponds to its current knowledge of the system, which includes:

- Y the winning bid utility for each target. For a radar i, $Y = (y_{ij})_{j \le |T|}$
- Z the identity of the winner for each target. For a radar i, $Z = (z_{ij})_{j \le |T|}$.

[2] Unsolvable in a reasonable time. Indeed, the presence of n boolean variables requires performing an enumeration, *i.e.* 2^n possibilities.

- S, which corresponds to a "timestamp" vector, it makes it possible to manage conflicts by making it possible to keep the track of the contacts between radars. For a radar i, $S_i = (s_{ik})_{k \leq |A|}$. It allows to select the most up-to-date information when there is a conflict among received information.

The CBBA algorithm has a performance guarantee of 50%, meaning that in the worst case, the global solution obtained is as good as half of the optimal solution. This algorithm is primarily intended for cases where planning is relevant, such as when agents are mobile robots and need to plan a route. However, in our case, we need to adapt this algorithm to make bids on a set of targets without taking the sequence order into account.

4.2 Adapting CBBA to Radars

In our case, for the allocation as the main radar, an additional vector will also be sent: the vector $E = (e_{ij})_{j \leq |T|}$, which groups the ellipses leading to the winning bids for each target. This allows for the calculation of intersections with the ellipses. To meet the Disminishing Marginal Gain (DMG) constraint required by CBBA, the utility of any new target being tracked needs to be reduced. To account for this constraint, we introduced the following bias:

$$c_{ij}^{CBBA} = \frac{c_{ij}}{|b_i|}$$

The algorithm operates in a closed loop and is executed at each time step. The agent first makes an allocation as the main radar, and then as an optional radar if it has remaining budget. Each allocation is made through the CBBA algorithm, which includes the two phases of the algorithm (auction and consensus) explained above. Therefore, it receives and sends information on its allocation as the main and optional radar at each time step. A radar does not consider the targets that it follows as the main radar in the list of targets that it can take as an optional radar.

To implement the interaction between radars, the ellipses sent by the radars are taken into account by the other radars to perform the utility calculation for the allocation as an optional radar. To follow as many targets as possible, when the agent computes its allocation as the main radar, it considers its budget as all of its remaining budget plus the budget allocated as an optional radar. If a new allocation as the main radar is possible, it deallocates the tasks as an optional radar with the lowest utility and performs a reset as described in the previous section.

In the static case, when all the radars have the same beliefs on the allocation, we say that the consensus is established. This means that a distributed allocation conflict-free could be found, which constitutes the end of the algorithm.

To account for the dynamic aspect of our problem, the auctions never stop and keep running until the end of the simulation. After a predetermined delay, the radar deletes the knowledge (which has become useless) that it had on the target and transmits the information. This reinitialization is used to reset the computed utility when the target moves away from a radar and the utility decreases. The general loop (repeated forever) is:

1. The radar enters the bidding phase as the main radar. To do so, it computes the uncertainty ellipses for each target and its utility function. It also applies the Kalman filters to all the targets it is already tracking.
2. If the radar has remaining radar time budget, it proceeds to the bidding phase as an optional radar on all the targets that are not already being tracked as the main radar (with its remaining budget).
3. The radar proceeds to the consensus phase as the primary radar. The vectors Y, Z, E, and S for the main radars are updated, and the vectors are sent to neighbors.

4. The radar then proceeds to the consensus phase as an optional radar. The vectors Y, Z, and S for the optional radars are updated, and the vectors are sent to neighbors.

5. The radar tracks the targets it has selected, possibly by applying its Kalman filter.

Note that the radar initiates tracks at each execution of the two phases of CBBA, which means that the algorithm does not have time to converge. In practice, this situation can generate conflicts, especially when a target is on the edge between two radars and becomes increasingly threatening. In such cases, each radar takes its decision based on a previous value of the utility of the others and its own current value. It considers that its bid wins the bet. Similarly, targets that become less threatening (i.e., with decreasing utility) may be tracked by none of the radars, with each radar considering that another radar has a better bid than itself.

Our approach solves the task allocation problem in an advantageous way. In the case where the radars remain in contact, even if the communication graph evolves during the mission, the algorithm keeps working as long as the connection graph of the radars remains path-connected. Finally, by carrying out two sequential auction runs, our algorithm also takes into account the possible overlapping of uncertainty ellipses. It thus generates an allocation that favors more precise tracking of targets when possible while trying to track as many targets as possible (depending on the radar capability).

Radars must also be able to differentiate between first-round allocation messages, which track as many targets as possible, and those that improve accuracy by generating an intersection of uncertainty ellipses. Each radar i tries to maximize the sum of the utilities corresponding to the targets that it tracks ($\sum_j x_{ij} \cdot c_{ij}$ for the main allocation and $\sum_{k,j} w_{ikj} \cdot c_{ikj}$ for the optional allocation), while taking into account the information received from the other radars, especially the bids made by them.

The implementation must include an additional target disambiguation mechanism to identify targets that can be tracked by multiple radars, including a plot merging algorithm to match the targets from different radars. This requires sending additional information, such as the estimated speed and position of the targets, to enable this operation to be carried out.

An illustration of the algorithm is represented on

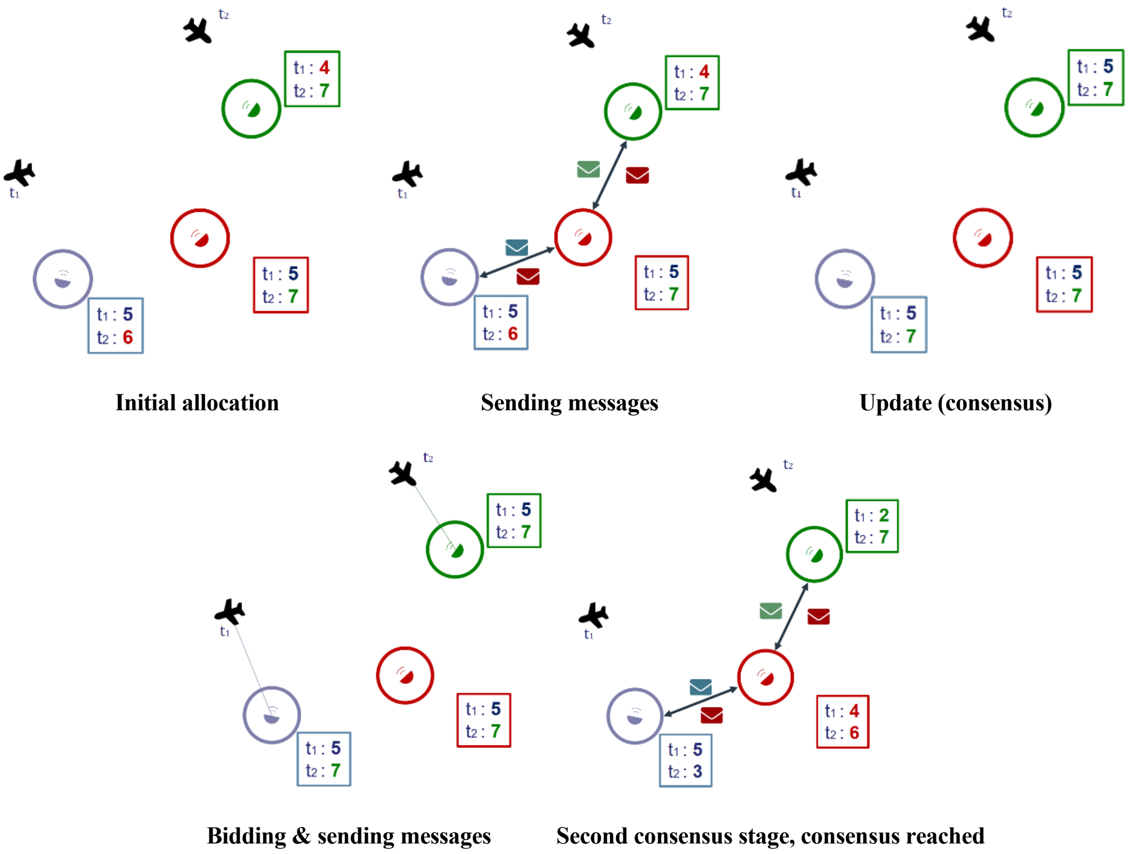

Figure 2: Illustration of our allocation method.

5.0 RESULTS

The implementation of our model has been performed using the MESA framework [6] and the Kalman filter package [4] on the same simulator used in [2]. We use 5 types of scenarios, which are similar but not identical to those used in [2]. The results are averaged over 10 scenarios for each graph.

- Non saturated well positioned radars (5 radars, 10 targets)
- Few saturated well positioned radars (3 radars, 12 targets)
- Several saturated well positioned radars (5 radars, 20 targets)
- Many saturated well positioned radars (8 radars, 30 targets)
- Saturated ill-positioned radars (4 radars, 20 targets)

A capture of the simulator is represented on the Figure 3. The red triangles represent the targets, the dark blue circles are the radars and the light blue circles represent their perception limits. The orange ellipses are the uncertainty areas around the targets. A green line represents a radar following a target as main radar. A violet line represents a radar following a target as optional radar.

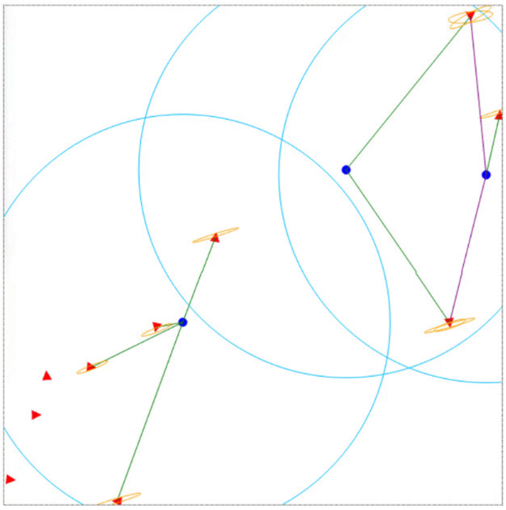

Figure 3: Capture of the MESA simulator.

In order to evaluate our work, we compare it to a centralized allocation performed with the Coin-OR Branch and Cut (CBC) tool [9]. The results are represented on Figure 4 and are averaged over 10 similar scenarios. The blue (resp. orange) curves correspond to the decentralized (resp. centralized) approach. The standard deviation is represented in light blue (resp. light orange).

The blue (or green) curves correspond to the decentralized (or centralized) approach. The average value of each of the simulations and the different scenarios for a certain "composition" of Targets and Radars are presented. Each of the "compositions" of Targets and Radars is represented on the x-axis as a tuple (Targets, Radars). This allows us to compare the performance of the centralized and decentralized approaches with equivalent configuration. The colors of the bars correspond to the decentralized (D) or centralized (C) values, with optional tracking represented in light colors when relevant. The standard error (in black) is available for each of the bars of the different graphs.

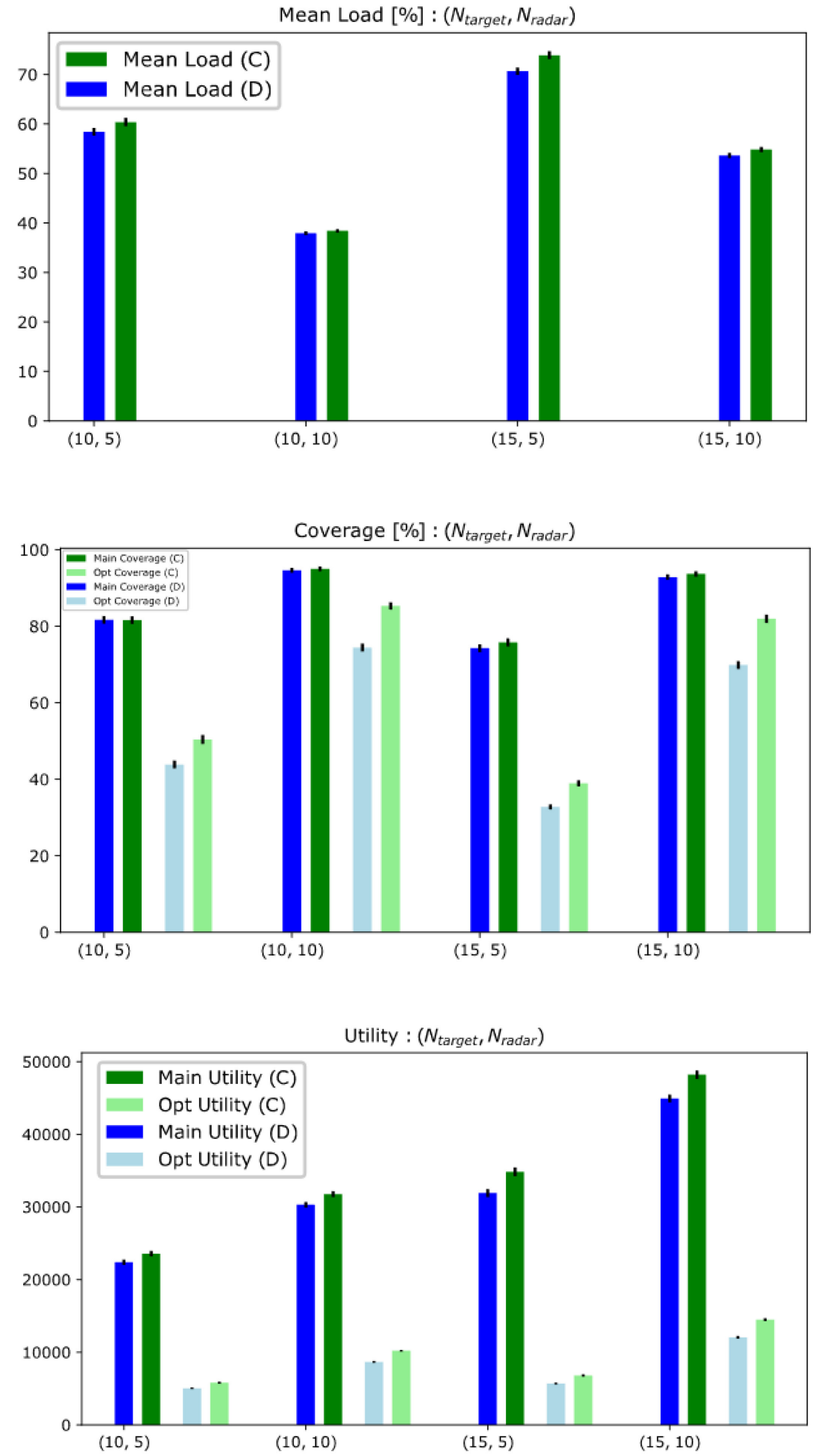

Figure 4: Utility, coverage and load on the reference scenarios.

Regarding the utility, as shown in the previous figure, the centralized approach obtains results superior to the decentralized approach. However for the decentralized approach we get a utility clearly higher than the theoretical 50% of the CBBA algorithm compared to the centralized approach.

Regarding the coverage, we notice that for a constant configuration, we obtain equivalent coverage in the "main" tracking. However, the coverage is weaker for the "optional" tracking. Regarding the average load, we observe that for the configurations studied, there is almost no difference between the centralized and decentralized approaches. While we would have liked the decentralized approach to have a much lower load than the centralized approach.

Overall, there are only very small differences between the centralized and decentralized approaches. This can be interpreted as a strength for the decentralized approach because the radar configuration could then be adaptive (one could suppose that the radars are not fixed but can move) but also resilient, i.e., the global system could continue to function normally if a connection is cut, which is not the case with a centralized approach since there is only one connection with the control center. Since the load is also more evenly distributed, it can be assumed that the decentralized approach can cope with a "surprise" attack without a total re-planning, which is not the case for the centralized approach. It should be noted that no experiment has been done with a higher number of radars since the constraint of the cost of purchasing a set of radars can be very limiting.

Overall, our approach performs almost as well as the centralized approach. When the radars are not saturated, it even performs better. In cases where the radars are saturated and numerous, a less complete search leads to targets not being tracked, and the centralized approach is better.

6.0 CONCLUSION

In this paper, we have proposed a novel approach for decentralized target allocation to a team of radars based on a fully decentralized auction algorithm, CBBA. We have demonstrated that this algorithm's results are comparable to centralized allocation when considering the intersection of uncertainty ellipses. Moreover, our approach performs better in cases where the radars are not too numerous and not saturated, but slightly worse when the radars are numerous and saturated.

Future works will focus on designing a more generic approach that can handle an arbitrary number of radars following the same target. Additionally, we aim to make our approach more dynamic by incorporating replanning approaches that have been proposed to improve CBBA [11]. We will also evaluate this approach in the setting imposed by our use-case.

7.0 REFERENCES

[1] S. S. Ponda, «Robust distributed planning strategies for autonomous multi-agent teams,» 2012.

[2] N. Nour, R. Belhaj-Soullami, C. L. R. Buron, A. Peres et F. Barbaresco, «Multi-Radar Tracking Optimization for Collaborative Combat,» chez *2021 21st International Radar Symposium (IRS)*, 2021.

[3] V. Lesser, C. L. Ortiz Jr et M. Tambe, Distributed sensor networks: A multiagent perspective, vol. 9, Springer Science & Business Media, 2003.

[4] M. Laaraiedh, «Implementation of Kalman filter with python language,» *arXiv preprint arXiv:1204.0375,* 2012.

[5] M. Krainin, B. An et V. Lesser, «An application of automated negotiation to distributed task allocation,» chez *2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07)*, 2007.

[6] J. Kazil, D. Masad et A. Crooks, «Utilizing python for agent-based modeling: the Mesa framework,» chez *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, 2020.

[7] B. Jia, K. D. Pham, E. Blasch, D. Shen et G. Chen, «Consensus-based auction algorithm for distributed sensor management in space object tracking,» chez *2017 IEEE Aerospace Conference*, 2017.

[8] M. Gini, «Multi-robot allocation of tasks with temporal and ordering constraints,» chez *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[9] J. Forrest et R. Lougee-Heimer, «CBC user guide,» chez *Emerging theory, methods, and applications*, INFORMS, 2005, p. 257–277.

[10] F. Deliang et R. Xiaomin, «A distributed sensor management algorithm based on auction,» *Procedia Computer Science,* vol. 107, p. 618–623, 2017.

[11] H.-L. Choi, L. Brunet et J. P. How, «Consensus-based decentralized auctions for robust task allocation,» *IEEE transactions on robotics,* vol. 25, p. 912–926, 2009.

[12] N. Buckman, H.-L. Choi et J. P. How, «Partial replanning for decentralized dynamic task allocation,» chez *AIAA Scitech 2019 Forum*, 2019.