USING MULTI-ARMED BANDITS FOR SEARCH APPLICATIONS IN A SPACE SURVEILLANCE RADAR NETWORK

Carolin Schwalm, Hans Schily, and Alexander Charlish

Department of Sensor Data and Information Fusion, Fraunhofer FKIE, 53343 Wachtberg, Germany,
Email: {carolin.schwalm, hans.schily, alexander.charlish}@fkie.fraunhofer.de

ABSTRACT

Before being able to track objects in space, a radar system first needs to operate in search mode in order to find objects of interest. A traditional search mode directs designated beams according to a predefined pattern by, for example, scanning the search volume from its upper left to its lower right corner. It does not, however, exploit possible acquired knowledge about tracks and object positions and hence might waste resources on areas that do not include any objects. In this paper, we propose modelling the search mode as a Multi-Armed Bandit (MAB) problem to allocate beams in the search volume while leveraging acquired knowledge. To achieve this, the single transmitter of a radar network is interpreted as the player of a MAB and the possible search beams as the arms of the slot machines that the player can choose from. The goal is to maximise the overall collected sum of rewards based on a predefined reward function, which represents the search effectiveness. We compare the performance of two standard MAB algorithms, namely the Epsilon-greedy and the Epsilon-decay algorithm, with the optimal decisions (in the sense of best in foresight) and the performance of a classic systematic search fence. We discuss to which extent the MAB approach is suitable for radar search applications and point out its chances and limitations.

Keywords: Multi-Armed Bandits; Space observation; Search Strategies.

1. INTRODUCTION

The purpose of optimizing a radar network operating in search mode is to detect as many objects as possible with sufficient precision to be able to track them. At the same time, the search function shall be executed efficiently, such that the radar has enough resources left to serve other functions like tracking. The unavoidable trade-off between different radar modes can be optimized using radar resource management techniques to make the allocation of resources as efficient as possible. Efficient allocation of sensor resources could mean that highly populated areas in a defined search space are covered more frequently, whereas fewer resources are spent for searching in sparsely populated areas. Some areas in the field of view of radar networks are more densely populated than others due to the fact that the tracks of satellite and debris objects orbiting the earth are often correlated [1]. Therefore, detected objects at a certain beam steering angle can be used to form knowledge on beam steering positions that are potentially more rewarding for searching. In other words, a radar operating in search mode gains situational knowledge with every measurement. Traditional search schemes of radar networks, however, do not directly exploit this knowledge. These search schemes scan the field of view systematically, e.g. from upper left to lower right positions in a predefined order, independent of the detections that are made over time.

This paper investigates a machine learning approach to exploit situational knowledge in order to adapt the search strategy and enable a more efficient allocation of resources. Concretely, we model the search mode as a Multi-Armed Bandit (MAB) problem. MAB problems address the dilemma of exploration versus exploitation: Exploration means that new situational information is gathered, while exploitation refers to taking advantage of the already acquired knowledge. In our case of a radar in search mode, we model the steering to unknown space areas as exploration, while the activation of beam positions that have previously led to detections can be seen as exploitation of the accumulated information. We implement and compare two different MAB algorithms, which provide decision policies in this exploration-vs-exploitation dilemma: the Epsilon-greedy and the Epsilon-decay algorithm. Our MAB model acts in an exemplary setting of a radar network consisting of one transmitter station and three receiver stations at fixed positions. For the implemented Single-Player MAB problem, it is important that only one Tx is considered, whereas the number of Rx stations can be arbitrary. The MAB algorithms can choose from

Proc. 2nd NEO and Debris Detection Conference, Darmstadt, Germany, 24-26 January 2023, published by the ESA Space Safety Programme Office
Ed. T. Flohrer, R. Moissl, F. Schmitz (http://conference.sdo.esoc.esa.int, February 2023)

a set of beam positions covering the search volume of interest. The evaluation of the MAB algorithms includes the comparison of their performance of with the optimal arm selection (in the sense of best in foresight) and the performance of a classic search fence, in which the search volume is scanned in a predefined regular order. In contrast to classic MAB problems, the rewards in some radar search scenarios can be heavily time-dependent. That is, highly rewarding arms appear when the game has already progressed, which can be hazardous for simple MAB algorithms. Therefore, we discuss to which extent the MAB approach is suitable for radar search applications and suggest possible modifications.

This paper is organized as follows. Section 2 provides the theoretical background of MAB theory and the adaptation of MAB algorithms to radar search. Also, we present the constellation and assumptions of the exemplary search scenario we set up for our simulations. Main simulation results are discussed in Section 3. Section 4 contains concluding remarks and an outlook to consecutive work.

2. MULTI-ARMED BANDITS: BACKGROUND AND IMPLEMENTATION

2.1. Multi-Armed Bandits: a Reinforcement Learning Problem

A MAB problem is a sequential decision problem whose name derives from a gambling scenario in which the player has to choose which arm in a row of non-identical slot machines to pull. In the classic MAB problem, each slot machine provides a pay-off drawn from a specific but unknown probability distribution. The gambler's intention is to maximize the overall sum of collected rewards. At each iteration, therefore, the gambler faces the dilemma to decide between pulling the lever of the slot machine with the currently highest expected reward or exploring a different machine to gain more information. The gambler makes this decision following a policy (i.e. an algorithm). As a metric for evaluating a policy, the regret ρ after K rounds is defined as the difference between the total expected reward of an optimal policy (i.e. consistently pulling the most rewarding arm) and the policy of interest:

$$\rho = \sum_{\kappa=1}^{K} r_{\kappa}^{*} - \sum_{\kappa=1}^{K} r_{\kappa}, \tag{1}$$

where r_k^* denotes the optimal and r_k the actual obtained reward in round k. More details on MAB theory can be found in [2], [3] and [4].

2.2. Modeling a Radar Search Scenario as a MAB Problem

This work applies the MAB model to a radar search scenario. Concretely, we simulate a radar network with one transmitter (Tx) and three receiver stations (Rx). Expressed in MAB terminology, the Tx acts as the player who, at each round, has to "pull a lever" in the sense of selecting one steering position $p(k)$ out of a set of predefined beam positions P. We assume that the Tx, as an electronically steered array, can switch almost instantly between beam positions in a negligible amount of time compared to the duration of each round of 0.1 seconds. Therefore, there are no restrictions with regards to the order of chosen positions $p(k),\ p(k+1), \ldots$: each position $p(k)$ at round k can technically be followed in round $k+1$ by any other position $p(k+1) \in P$. After having selected a beam position, the player receives a reward r_k for this round. In our implementation of the MAB problem, r_k is either the overall sum of object detections that the three Rx deliver or the number of uniquely detected objects. If, for instance, either n detections or n newly detected objects are registered at round k, the Tx updates its knowledge by increasing the rating of the chosen Tx beam position $p(k)$ by n. In order to evaluate the regret ρ, the optimal reward r_k^* is also calculated at each round in the simulation. r_k^* corresponds to either the number of detections or unique objects n^* that an omniscient player could have achieved if he had known the trajectories of the targets in advance and hence had chosen the beam position accordingly. This optimal behavior can, naturally, not be achieved by any bandit with limited knowledge. ρ, however, is a useful metric for comparing different MAB algorithms. The smaller ρ, the closer the accumulated obtained reward to the reward obtained under hindsight knowledge, hence the better the performance of the algorithm.

Many different MAB strategies and algorithms exist with varying complexity, game constellations and assumptions. This work investigates the feasibility of modeling a radar search scenario as a MAB problem. We therefore keep the focus on standard, simple algorithms for addressing the MAB problem. Concretely, we compare different implementations of greedily behaving algorithms. Greedy algorithms select the best rated lever (based on previously collected rewards) in the exploitation phase and select a lever randomly (in our case out of a uniform distribution) in the exploration phase, cf. [2]. The proportions of time in which either the exploitation or the exploration phase is active depends on the parameterization of the algorithms. Specifically, we implement a so-called Epsilon-greedy and an Epsilon-decay algorithm. The Epsilon-greedy algorithm is in exploitation mode during $1-\epsilon$ of the rounds and in exploration mode for a proportion of ϵ. In the Epsilon-decay algorithm, the value of ϵ decreases in time. Thus, the player acts ex-

ploratory at the start and increasingly exploitative as the game progresses. Pseudo-code descriptions of both simple algorithms are given in the following.

```
Epsilon-greedy algorithm
  p = random() % from U_[0,1)
  if p < ε
    pull random action
  else
    pull best rated arm

Epsilon-decay algorithm
  p = random() % from U_[0,1)
  if p < ε
    pull random action
    ε = ε * decay_factor(time)
  else
    pull best rated arm
```

U denotes the uniform distribution. In our implementation of the Epsilon-greedy algorithm, we start with $\epsilon = 1$ at $k = 0$ and update the epsilon at each round with $\epsilon = \frac{1}{1+k/10}$, where k denotes the round being played.

The arrangement of the beams available to the MAB player is depicted in Figure 1. The fixed left-to-right pattern search fence used for comparison selects consecutive beams, moving row-wise from the upper left one to the lower right one.

2.3. Simulated Scenario

This paper considers a radar network with one Tx and three Rx being between approximately 24 km and 34 km apart from the Tx. Note that in this simulation, we model a Single-Player MAB problem, therefore any configuration using a single Tx would be suitable, including the monostatic and bistatic case with a single Rx. Figure 2 shows the sensor locations used for this simulation. All apertures have a fixed position in an Earth-centered, Earth-fixed (ECEF) frame that is configured according to the WGS84 conventions. The implementation is based on the Orekit space dynamics library [5] to compute orbital dynamics and coordinate transformations. All antennas are tilted 30 degrees northwards with respect to the local zenith of the station.

The simulation uses publicly available data from [6] to create realistic target trajectories from all known two-line element sets (TLE). The trajectories are filtered to contain only objects up to an altitude of 3000 km. Furthermore, the scenario begins on Feb. 22nd 2022 at 10:25 am and ends 4 minutes later, while any beam chosen by the MAB is active for 0.1 seconds. The Rx schedule beams independently from each other in order to completely cover any active Tx beam. For simplicity, a detection is declared if an object is caught in a Tx and Rx beam simultaneously. Since the MABs act randomly during the exploration phase, four Monte Carlo runs of the simulation are averaged for assessing their performance. Figure 3 illustrates the trajectories of the objects crossing the field of view of the Tx.

3. SIMULATION RESULTS

In order to evaluate the performance of the implemented MAB algorithms, we firstly define the reward as the overall sum of object detections, independent of how many different objects (in the sense of unique object identifiers) have been detected. Figure 6 shows a comparison of two different bandit algorithms with the left-to-right search fence and additionally with a policy selecting beams randomly. The sum of rewards (in the sense of detections), averaged over 4 total simulations, are plotted over the rounds played.

Clearly, neither the bandit algorithms nor the left-to-right or total-random schemes collect maximum rewards as they all act under limited knowledge. However, all bandit algorithms collect more detections than the left-to-right scan or the random policy. Regarding the total reward at the end of the simulation, the Epsilon-greedy algorithm with $\epsilon = 0.2$ performs best, collecting $1280/3189 \approx 40.14$ % of the detections. Next is the Epsilon-decay algorithm collecting approximately 37.28 % of the overall reward. The Epsilon-greedy algorithm with $\epsilon = 0.4$ achieves approximately 24.24 % of the maximum detections and the Epsilon-greedy algorithm with $\epsilon = 0.9$ detects 25.24 %. The left-to-right search fence and the random policy detect 26.32 % of all possible detections.

Figures 4 and 5 show the trajectories of the objects passing the Tx beams. The beams are colored with respect to the normalized number of detections collected in each beam at the end of the simulation. The normalization is made on the respective maximum of the number of detections. Figure 4 shows the distribution of the maximum reward. This resembles the knowledge an omniscient MAB would accumulate during a game. In comparison, Figure 5 shows the knowledge of the bandit with the Epsilon-greedy algorithm with $\epsilon = 0.2$ at the end of a simulation and averaged over four simulations. Evaluating Figure 5 reveals that the bandit algorithm correctly identifies a frequented beam (in the 3rd row, second bin from the left) and uses it to generate a high amount of detections during the exploitation phase. However, it does not identify all opportunities to collect many detections. Since it greedily exploits its knowledge, it does not diversify and fails to uncover the possibility of collecting detections through sampling e.g. the top right beams.

In addition, considering Figure 6, we can see that after approximately 2000 played rounds, the slope

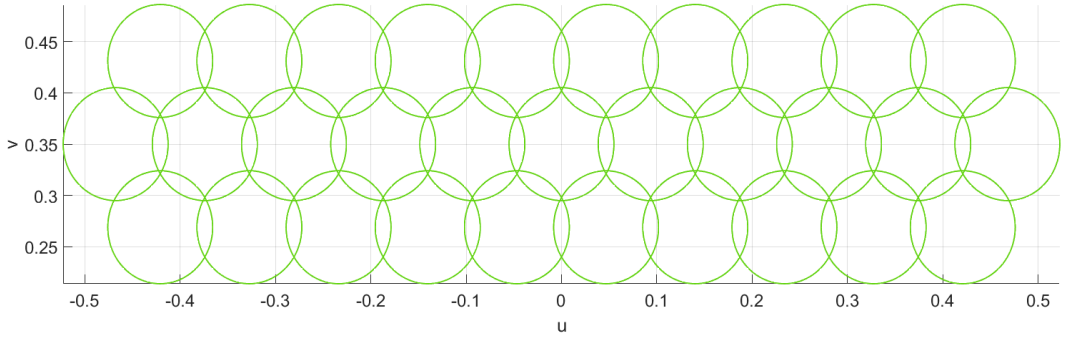

Figure 1. Beam positions as seen from the Tx in Tx antenna coordinates (u, v).

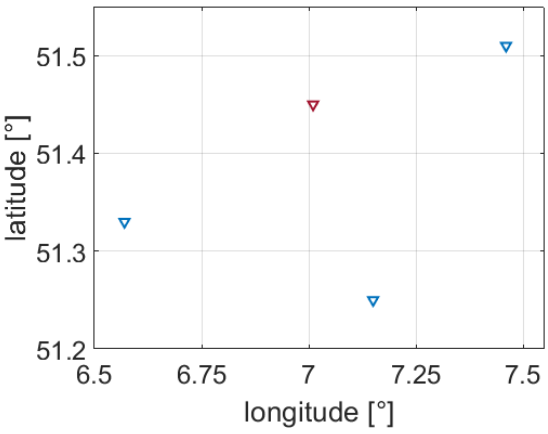

Figure 2. Radar network configuration with one Tx (red) and three Rx (blue).

of the available maximum reward increases, indicating that there is a new optimal beam to choose. The Epsilon-greedy bandit, however, is designed such that it sticks to its most promising beam (in that case, the third bin from left, first row, see Figure 5) for on average 80 % of the rounds as the simulation advances. In other words, the bandit algorithm gets stuck in a local maximum if the global maximum becomes apparent when the game has already progressed and the knowledge the player collects in its exploration phases cannot outweigh the knowledge basis of the exploitation phase. Consequently, the exploitation strategy of these simple bandits does not match the underlying time variant distribution of reward.

So far, we based our comparisons on the total number of detections. In other words, the reward that the MAB uses for building its knowledge is based on detections and it ignores the uniqueness of detected objects. One single object may, for instance, generate a lot of hits when it passes comparably slowly through one beam. The bandit would, in that case, increase the rating of this beam (and, hence, choose it more often in future rounds), but not necessarily be able to detect yet unseen objects.

In order to be able to judge the feasibility of bandit algorithms in radar search scenarios, we further compare the uniqueness of the detected objects, i.e. how many different objects are registered by following the MAB algorithms.

Figure 7 shows that all bandit algorithms detect fewer unique objects than the left-to-right scheme of activating beams. This clearly shows how optimizing for detections comes at a significant cost. The more effort the MAB spends in the exploitation phase (which is on average 80 % of the rounds for the Epsilon-greedy algorithm with $\epsilon = 0.2$ and exponentially grows as the game progresses for the Epsilon-decay algorithm), the fewer different objects are detected. This of course is due to the fact that the metric on which the MAB bases its knowledge is purely detection-based. Detections of new unknown objects are not directly rewarded; only the accumulation of detections brings benefits to the MAB.

The scenario is designed such that the left-to-right scheme is difficult to beat by default because it scans the beams fast enough so that few objects can cross the beam pattern from north to south without being caught by at least one beam. That way, it is highly probable that most objects are detected by the left-to-right beam activation pattern at least once, unless they pass by at the very edges of the search space. However, the bandit algorithms, especially the Epsilon-decay algorithm, stick to promising beams by design; thus, they have a higher probability of missing new and unknown objects.

Figure 8 shows that by changing the metric for generating rewards from the sum of detections to the uniqueness of detected objects, the Epsilon-greedy algorithm with $\epsilon = 0.2$ is able to generate more detections than in the configuration presented in Figure 6. This is due to the fact that it now converges in its exploitation phase to a beam in the upper right corner, which is populated by many different objects and therefore also generates many detections when selected, cf. Figure 9.

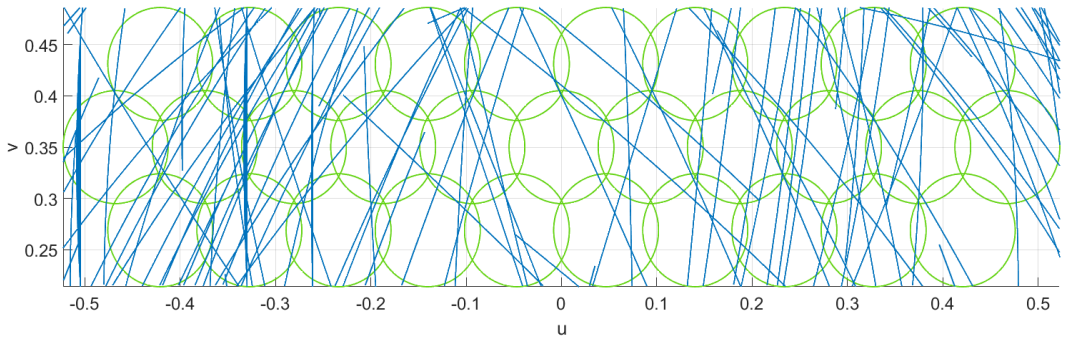

Figure 3. Beam positions as in Figure 1 with the trajectories of crossing objects.

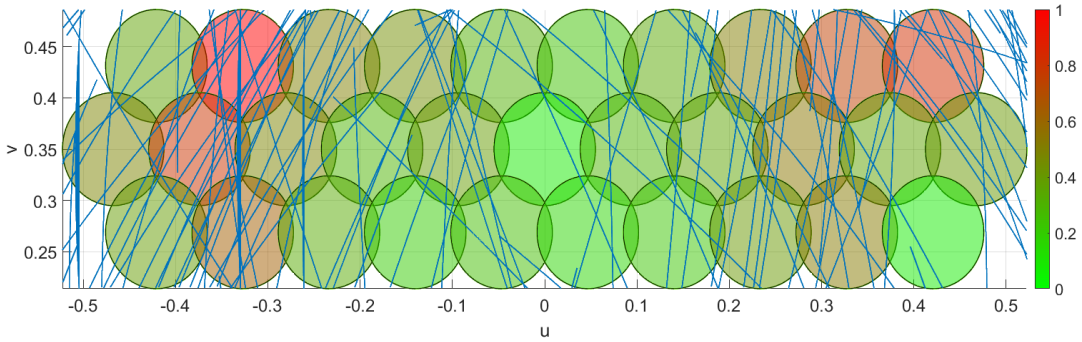

Figure 4. Normalized heatmap of the maximum possible detections for each beam. The blue lines mark the trajectories of passing objects.

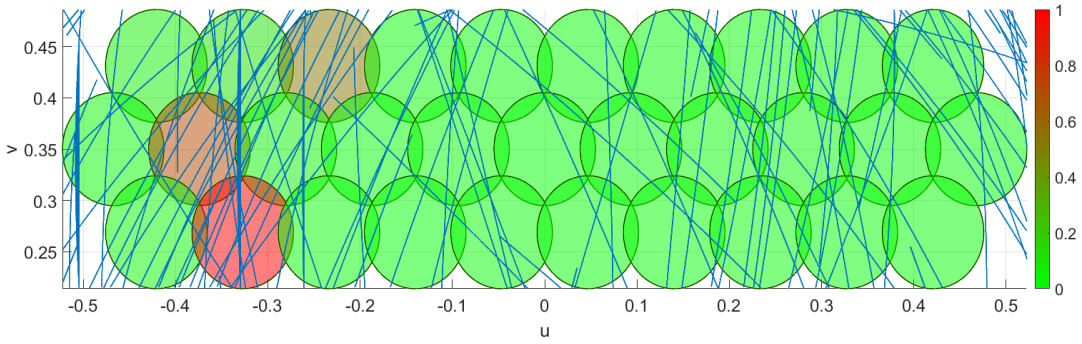

Figure 5. Normalized heatmap of the accumulated detections for each beam of the Epsilon-greedy bandit, $\epsilon = 0.2$, if rewarded based on number of detections. The blue lines mark the trajectories of passing objects, the same as in Figure 4.

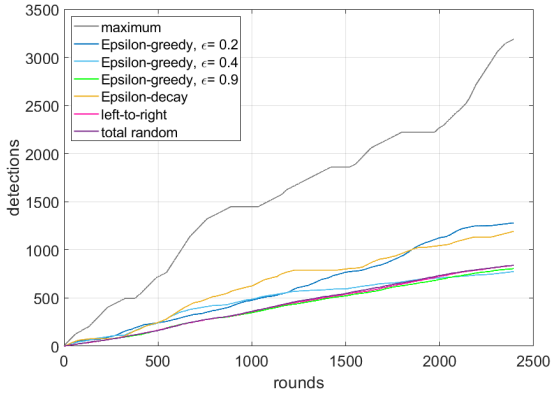

Figure 6. Comparison of the performance of different search strategies, measured on the basis of detections (averaged over 4 Monte Carlo runs).

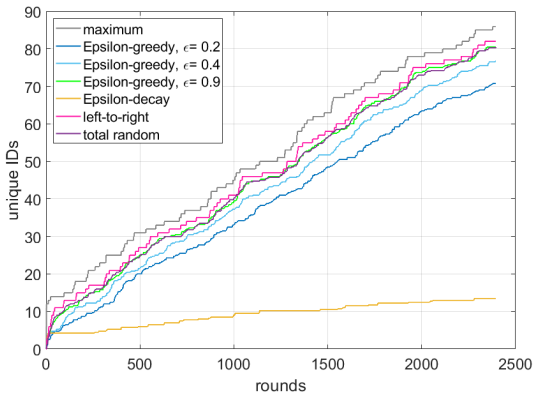

Figure 7. Comparison of the performance of different search strategies, measured on the basis of unique object IDs (averaged over 4 Monte Carlo runs).

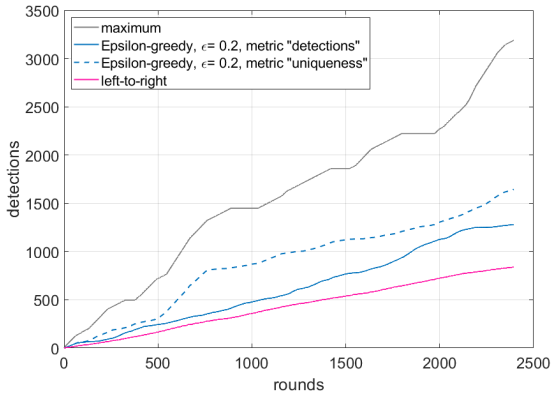

Figure 8. Comparison of the performance if additionally the reward metric is based on uniqueness of detected objects (averaged over 4 Monte Carlo runs).

Therefore, we can identify one possible advantage of the Epsilon-greedy algorithm with $\epsilon = 0.2$ if it is rewarded based on the diversity of detected objects: It identifies a beam with a local maximum of the diversity of objects and generates more detections than the left-to-right beam selection pattern. To illustrate this assumption, we build the quotient $q(k)$ as the number of overall detections over number of discovered unique objects:

$$q(k) = \frac{\sum_{\kappa=1}^{k} r_{\kappa,\text{detections}}}{\sum_{\kappa=1}^{t} r_{\kappa,\text{uniqueness}}}. \tag{2}$$

The distribution of q is shown in Figure 10. As can be seen, the Epsilon-greedy MAB algorithm with $\epsilon = 0.2$ generates more detections per detected object than the left-to-right scheme consistently over the duration of the simulation. However, as mentioned before, this comes at the price of missing some objects that could be detected by spending the available resources on a different beam activation pattern.

4. CONCLUSION

In general, the search task of a radar network can be expressed in MAB terminology. The corresponding player is, in our case, the single Tx, having to choose one beam ("lever") at each round to activate ("play") and therefore generate detections from objects crossing its field of view. We implement multiple MAB algorithms of type Epsilon-greedy and Epsilon-decay, which act exploratory during a certain part of the rounds and exploitative (in the sense of choosing the presumed most-rewarding beam) for the rest of the time. If we reward the player based on accumulated detections, the implemented bandit algorithms quickly identify the most rewarding beams in terms of the number of overall detections. However, there is one drawback to this strategy: Rewarding only detections lets the MAB be oblivious of the usefulness of detecting new objects.

If the rewards are based on the diversity of detected objects, we can identify an advantage of the Epsilon-greedy algorithm with a short exploration phase of $\epsilon = 0.2$. It quickly identifies a beam with diverse objects and generates more detections from it as it chooses this beam more often than a left-to-right systematic beam activation scheme. This can be advantageous for a reliable initialization of object tracks.

One general drawback of simple bandit algorithms becomes apparent if highly rewarding arms appear when the game has already progressed, as it is the case when targets enter after multiple rounds have already been played. The underlying optimal beam activation pattern is, in this case, time dependent. The player might get stuck in a local maximum, with an even more drastic effect for algorithms with an decaying exploration phase like the Epsilon-decay algorithm.

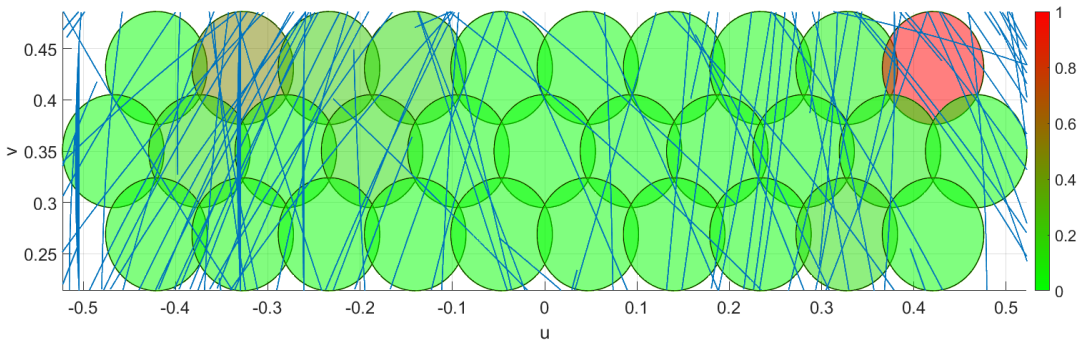

Figure 9. Normalized heatmap of the accumulated detections for each beam of the Epsilon-greedy bandit, $\epsilon = 0.2$, if rewarded based on diversity of discovered objects.

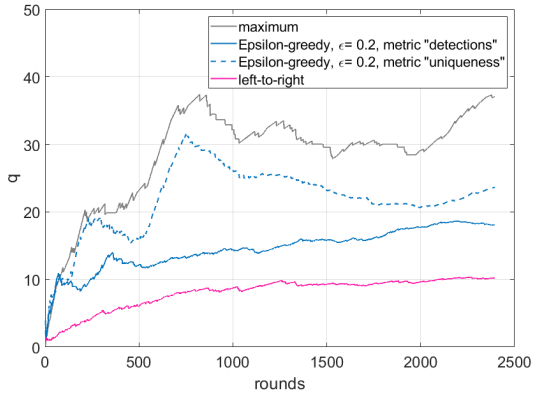

Figure 10. Quotient q as defined in Equation (2).

In conclusion, a MAB algorithm, if properly designed, is advantageous to detect initially highly populated beams from a fixed grid of beams and generates more detections than a left-to-right beam activation pattern or a random policy. This mimics a Track-While-Scan behavior of a radar and should lead to a more accurate track initialization and a reduced tracking error for the discovered space objects.

However, if the underlying distribution of detectable objects is heavily time varying, relying on greedy agents does not suffice. Instead of a purely greedy exploitation of the available knowledge, a more complex version should be implemented that broadens the exploitation pattern. In addition, the MAB could include a-priori knowledge or learn to forget targets and, therefore, more quickly adapt to a changing scenario.

The knowledge of the MAB could be especially useful if the available sensor resources are reduced because the radar has to serve other functions like tracking. Then, a MAB has the potential to outperform the systematic left-to-right pattern. Similarly, the MAB could be applied to a situation in which the area of interest for the search is too large to be served systematically. In that situation, the left-to-right pattern would also lack resources and presumably miss more objects.

Another area where MAB have potential useful applications is if multiple Tx have to search some volume in space collaboratively. These multi-player MAB could be designed such that the MAB benefit from each other's knowledge and automatically coordinate their beams to a rewarding configuration. Ideally, a MAB algorithm would converge to a known reliable activation pattern for the case of fully available resources, but smartly exploits its knowledge in case of reduced resources.

ACKNOWLEDGMENTS

The project underlying the presented results was realized with funds from the German Federal Ministry of Economic Affairs and Climate Action under the grant with code 50LZ2005. The author is responsible for the content of this publication.

REFERENCES

1. Schildknecht, T. (2007). Optical surveys for space debris. The Astronomy and Astrophysics Review 14.1, 41–111.
2. Slivkins, A. (2019). Introduction to multi-armed bandits. Foundations and Trends in Machine Learning 12.1-2, 1–286.
3. Auer, P., Cesa-Bianchi, N. and Fischer, P (2002). Finite-time analysis of the multiarmed bandit problem. Machine learning 47.2, 235–256.
4. Robbins, Herbert (1952). Some aspects of the sequential design of experiments. Bulletin of the American Mathematical Society 58.5, 527–535.
5. L. Maisonobe et al. (2010). Orekit: An open source library for operational flight dynamics applications. 4th International Conference on Astrodynamics Tools and Techniques, 3–6.
6. https://celestrak.org/