Multi-Radar Tracking Optimization for Collaborative Combat

Nouredine Nour, Reda Belhaj-Soullami, Cédric Buron, Alain Peres, Frédéric Barbaresco

Multi-Radar Tracking Optimization for Collaborative Combat

Nouredine Nour[1], Reda Belhaj-Soullami[1], Cédric L.R. Buron[2], Alain Peres[3], and Frédéric Barbaresco[3]

[1] NukkAI, Paris, France (www.nukk.ai),, email: {nnour, rbelhaj-soullami}@nukk.ai
[2] Thales Research & Technology, 1 av. Augustin Fresnel, Palaiseau France, email: cedric.buron@thalesgroup.com
[3]Thales Land & Air Systems, 3 Avenue Charles Lindbergh, 94150 Rungis, France , email: {alain.peres, frederic.barbaresco}@thalesgroup.com

October 19, 2020

Abstract

Smart Grids of collaborative netted radars accelerate kill chains through more efficient cross-cueing over centralized command and control. In this paper, we propose two novel reward-based learning approaches to decentralized netted radar coordination based on black-box optimization and Reinforcement Learning (RL). To make the RL approach tractable, we use a simplification of the problem that we proved to be equivalent to the initial formulation. We apply these techniques on a simulation where radars can follow multiple targets at the same time and show they can learn implicit cooperation by comparing them to a greedy baseline.

Keywords: Netted sensors, Reinforcement Learning, Actor Critic, Evolutionary Algorithms, Multi-Agent Systems

1 Introduction

Despite great interest in recent research, in particular in China [1, 2] micromanagement of sensors by centralized command and control drives possible inefficiencies and risk into operations. Tactical decision making and execution by headquarters usually fail to achieve the speed necessary to meet rapid changes. Collaborative radars with C2 must provide decision superiority despite the attempts of an adversary to disrupt OODA cycles at all level of operations. Artificial intelligence can make a contribution for the purposes of coordinated conduct of the action, by improving the response time to threats and optimizing the allocation and the distribution of tasks within elementary smart radars.

In order to address this problem, Thales and the private research lab NukkAI have been collaborating to introduce novel approaches for netted radars. Thales provided the simulation modeling the multi-radar target allocation problem and NukkAI proposed two novel reward-based learning approaches for the problem.

In this paper, we present these two approaches: Evolutionary Single-Target Ordering (ESTO), which is based on evolution strategies and an RL approach based on Actor-Critic methods. To make the RL method tractable in practice, we introduce a simplification of the problem that we prove to be equivalent to solving the initial formulation. We evaluate our solutions on diverse scenarios of the aforementioned simulation. By comparing them to a greedy baseline, we show

that our algorithms can learn implicit collaboration. The paper is organized as follows: section 2 introduces the related works. The problem is formalized in section 3. In section 4, we describe the proposed approaches. Section 5 presents the results of our simulations. section 6 concludes the paper.

2 Related works

Decentralized target allocation in a radar network has gained a lot of interest recently [3]. For this problem, resolution through non-cooperative game formalism [4] reaches good performance, but only considers mono-target allocation. Bundle auctions algorithm [5] overcomes this limitation; still, none of these approaches are able to model the improvement provided by multiple radars tracking the same target. Another suitable method is reward-based machine learning, that can either take the form of evolutionary computation [6] or reinforcement learning (RL). Recent successes in multi-agent RL were obtained by adapting mono-agent deep RL methods to the multi-agent case, most of them based on policy gradient approaches [7] with a centralized learning and decentralized execution framework [8, 9, 10]. In the policy gradient algorithm family, actor-critic methods [11] relying on a centralized critic have empirically proven to be effective in the cooperative multi-agent partially observable case [9]. However, the size of the action space requires to adapt these approaches for our problem.

3 Problem statement

In this paper, we consider that each radar can track a set of targets that move in a 2D space (for sake of simplicity, elevation is ignored). The targets are tracked with an uncertainty ellipse, computed using a linear Kalman model. We assume that the radars have a constant Signal-to-Noise Ratio (SNR) on target (adaptive waveform to preserve constant SNR on target), can communicate without limitations, have a limited field of view (FOV) and a budget representing the energy they can spend for tracking capabilities. Their Search mode are not simulated but taken into accoun the the constrained time budget for active track modes.

Let n be the number of radars and m the number of targets. In our model, an action of a radar is the choice of the set of targets to track. If multiple radars track the same target, the **uncertainty area** is the superposition of the uncertainty ellipses. We define a utility function $\mathcal{U}$ measuring the global performance of the system. Let l_i^j be the elementary radar i budget needed to track target j, L_i the budget of radar i, $\mathcal{E}_i^j$ the uncertainty ellipse on target j for radar i and $S(\mathcal{E}_j^i)$ the area of $\mathcal{E}_j^i$ (improvement is possible by considering covariance matrix of trackers). The problem can be expressed as a constraint optimization problem:

$$\text{maximize } \mathcal{U} = \frac{1}{m}\sum_{j=1}^{m} \exp\left[-S\left(\bigcap_{i=1}^{n} \mathcal{E}_i^j\right)\right] \text{ such that: } \forall i \leq n, \sum_{j=1}^{m} l_i^j \leq L_i \tag{1}$$

4 Task allocation in a radar network

4.1 Evolutionary Single-Target Ordering (ESTO)

ESTO is a centralized training with decentralized execution black-box optimization method. Based on contextual elements, agents define a preference score for each target. They then choose the targets to track greedily based on this score, until their budget is met. The preference score is computed by a parametrized function optimized to maximize the utility using the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [12]. CMA-ES optimizes the parameters by sampling them from a normal distribution and updating both the mean and the covariance matrix based on the value of the utility function. We modelized ESTO's preference score function with a linear model with 9 input features including information on the target, the radar, and the position

of the other radars. We also propose a variant of ESTO, called ESTO-M which takes into account 2 additional features based on inter-radar communication: the estimated target utility and the evolution of the estimated utility from the previous step.

4.2 The Reinforcement Learning approach

Dec-POMDP formulation. Collaborative multi-agent reinforcement learning typically relies on the formalization of the problem as a Dec-POMDP [13]. It is defined as a tuple $\langle D, S, A, P, R, \Omega, O\rangle$, where D is the set of agents ($|D| = n$), S is the state space, $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$ is the joint action set, P is the transition function, R is the reward function, Ω is the set of observations and O is the observation function. A state can be described as a tuple containing the true position and velocity of each target, the position of the other radars, and the Kalman filter parameters for each radar-target pair. The transition dynamics include the update of the Kalman filters. The problem is not fully observable: the radars approximate the position and speed of the targets and can't access to the Kalman filters of the others. Radars rely on a $m \cdot n_f$ real-valued vector, with n_f the number of features: estimated position, speed, etc.

Our approach is based on centralized learning and decentralized execution. Although this approach may lead to stationarity issues, it is widely used in practice and yields good results in multi-agent RL [14, 9]. When applying policy, the probability of tracking targets beyond FOV is set to 0, the probabilities of remaining targets are updated accordingly. Agents share the same network but its input values depend on the radar and the targets.

In this setting, the size of the action space corresponds to all the possible allocations of sets of targets to each agent $\forall i \leq n, \mathcal{A}_i = \mathcal{P}(\llbracket 1, m \rrbracket)$ and $|\mathcal{A}_i| = 2^m$, *i.e.* the powerset of all targets. To tackle this issue, we propose a new formalization of the problem where the radars choose the target sequentially, and prove the equivalence between the solutions of the two formalisms.

Definition 1 (Sequential choice Dec-POMDP)**.** Let $M = \langle D, S, A, P, R, \Omega, O\rangle$ be a Dec-POMDP for our problem. Let $M' = \langle D, S', A', P', R', \Omega, O'\rangle$ with $S' = S \cup \{s \otimes \varepsilon | s \in S, \varepsilon \in \mathcal{P}(\llbracket 1, m \rrbracket \cup \dagger)^n\}$ where ε is the set of targets tracked by each agent and the symbol $\dagger$ means that its allocation is finished; $s \otimes \varepsilon$ is a notation for the new (couple) state in S'. In a state $s \otimes \varepsilon \in S'$, the set of allowed actions of agent i is $(\llbracket 1, m \rrbracket \cup \{\dagger\}) \setminus \varepsilon_i$. The observation, state-transition and reward functions are defined as:

$$\forall a \in A', \forall (\varepsilon, \varepsilon') \in \left(\mathcal{P}(\llbracket 1, m \rrbracket)^n\right)^2, \forall (s, s') \in S^2, O'(s \otimes \varepsilon, s' \otimes \varepsilon) = O(s, s')$$

$$P'(s \otimes \varepsilon, a, s' \otimes \varepsilon') = \begin{cases} P(s, \varepsilon, s') \text{ if } a = (\dagger, \dots, \dagger), \varepsilon' = \varnothing \\ 1 \text{ if } s = s', \varepsilon'_j = \varepsilon_j \cup \{a_j\} \forall j \leq n \\ 0 \text{ else} \end{cases}$$

$$R'(s \otimes \varepsilon, a, s' \otimes \varepsilon') = \begin{cases} R(s, \varepsilon, s') \text{ if } a = (\dagger, \dots, \dagger), \varepsilon' = \varnothing \\ 0 \text{ else} \end{cases}$$

This new Dec-POMDP can be solved much more easily than the initial one. We now look for a solution of the initial Dec-POMDP from the sequential choice Dec-POMDP. In the rest of the article, we denote by V (resp. V') the averaged state value function in M, (resp. M'): $V_\pi(\rho) = \mathbb{E}_{a_t \sim \pi, s_0 \sim \rho}\left(\sum_{t=1}^{T} R(s_t, a_t, s'_t)\right)$. For space reasons, only sketches of the lemma proofs are provided.

Definition 2 (Policy transposition)**.** We define the policy transition function ϕ from the set of policies in M' to the set of policies in M as

$$\forall j \in \llbracket 1, n \rrbracket, \phi_j(\pi')(\varepsilon|\omega) = \sum_{\{i_k\} = \varepsilon} \pi'_j(i_1|\omega)\pi'_j(i_2|\omega \otimes \{i_1\}) \dots \pi'_j(\dagger|s \otimes \{i_1, \dots, i_p\})$$

Lemma 1 (Value equivalence)**.** *Let $\pi = \phi(\pi')$. Let ρ be a probability distribution on S and π' a policy on M'. Then $V'_{\pi'}(\rho) = V_{\phi(\pi)}(\rho)$.*

Proof (sketch). By definition 1, the result holds iff $\forall (s, \varepsilon, s') \in S \times \mathcal{P}([\![1, m]\!])^n \times S$,

$$\rho(s)\pi(\varepsilon|\omega)P(s, \varepsilon, s') = \sum_{\{i_k\}=\varepsilon} \rho(s)P'(s, i_1, s \otimes i_1)\pi'(i_1|\omega) \dots P'(s \otimes \varepsilon, \dagger, s')\pi'(\dagger|\omega \otimes \varepsilon)$$

By using definition 1 the equation simplifies exactly to the one of definition 2. □

Lemma 2 (Surjectivity)**.** *The mapping ϕ is surjective. Let π be a policy on M. Then $\pi = \phi(\pi')$ with π' defined the following way (ω is omitted):*

$$\begin{aligned} &\forall k \in [\![1, n]\!], \forall j \in [\![1, m]\!] \setminus \varepsilon_k, \pi'_k(\dagger|\varepsilon_k) = \frac{1}{N_{\varepsilon_k}} \pi_k(\varepsilon_k) \frac{\pi_k(A)|\varepsilon_k|!}{|A| \dots (|A| - |\varepsilon_k|)} \\ &\text{and } \pi'_k(j|\varepsilon_k) = \frac{1}{N_{\varepsilon_k}} \sum_{\substack{A \subset [\![1,m]\!] \\ \varepsilon \subset A \\ j \in A}} \quad \text{with } N_{\varepsilon_k} = \sum_{\substack{A \subset [\![1,m]\!] \\ \varepsilon_k \subset A}} \frac{\pi_k(A)|\varepsilon_k|!}{|A| \dots (|A| - |\varepsilon_k| + 1)} \end{aligned} \tag{2}$$

Proof (sketch). First, we verify that $\sum_{j \in [\![1,m]\!] \setminus \varepsilon_k} \pi'_k(j|\varepsilon_k) = 1 - \pi'_k(\dagger|\varepsilon_k)$. Then we show that $\pi_k = \phi(\pi'_k)$. Let $\varepsilon = (i_1, \dots, i_p)$, and $(j_1, \dots, j_p)$ an arbitrary permutation of ε. Let $\varepsilon_l = (j_1, \dots, j_l)$. Then, for all $l \in [\![0, p-1]\!], \pi'_k(j_{l+1}|\varepsilon_l) = \frac{N_{\varepsilon_{l+1}}}{N_{\varepsilon_l}(l+1)}$ with $\varepsilon_0 = \varnothing$. The product then simplifies to $\pi'_k(j_1)\pi'_k(j_2|j_1) \dots \pi'_k(\dagger|\{j_1, \dots, j_p\}) = \frac{\pi_k(\varepsilon_k)}{p!}$. Summing among all $p!$ permutations of $[\![1, p]\!]$, we verify that $\pi_k = \phi(\pi'_k)$. □

Theorem 1. *Let π'_* be an optimal policy in M', then $\pi_* = \phi(\pi'_*)$ is an optimal policy in M.*

Proof. This follows directly from the surjectivity and value equivalence lemmas. □

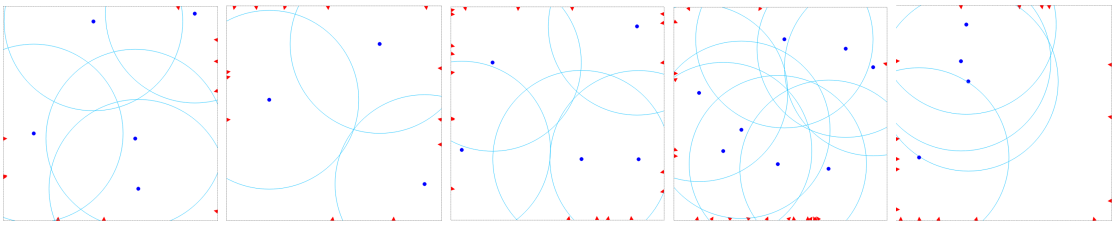
Actor-Critic methods. To find a policy that maximizes the expected average reward, we used Proximal Policy Optimization (PPO) [15], a variant of the actor-critic algorithm. Although the algorithm is only proved for MDPs, the use of a centralized critic has proven to be efficient in simple partially observable multi-agent settings [9]. PPO relies on an actor that plays episodes according to a parametrized policy and a critic that learns the state value function. After each batch of played episodes, the parameters of the two networks are updated according to the surrogate loss as in [15].

Neural network architecture. The critic neural network architecture is a standard multi-layer perceptron. Regarding the actor, the first layer consists of n_f neurons : an input tensor of size (m, n_f) is passed to the network instead of a first layer of $n_f \cdot m$ neurons. The network consists of a **feature extractor** of two layers reducing the number of features from 23 to 6 and a **feature aggregator** consisting of two linear models T and O, that represent respectively the contribution of the target itself, and the interest of the other targets. Intuitively, training at individual target level allows better feature extraction and generalization than a dense, fully connected architecture. Moreover, it allows to ensure full symmetry of the weights. However, this comes at the cost of expressiveness, as we use a special form of architecture for our actor. Let f be the extracted feature matrix : f_i is the extracted features for target i. We compute the score w_i of target i as $w_i = T(f_i) + \frac{1}{m-1} \sum_{j=1, i \neq j}^{m} O(f_j)$. The process is converted to a probability using a softmax activation function and can be represented as:

3@23 feat. → feature extractor → 3@6 feat. → feature aggregator → 3 scores

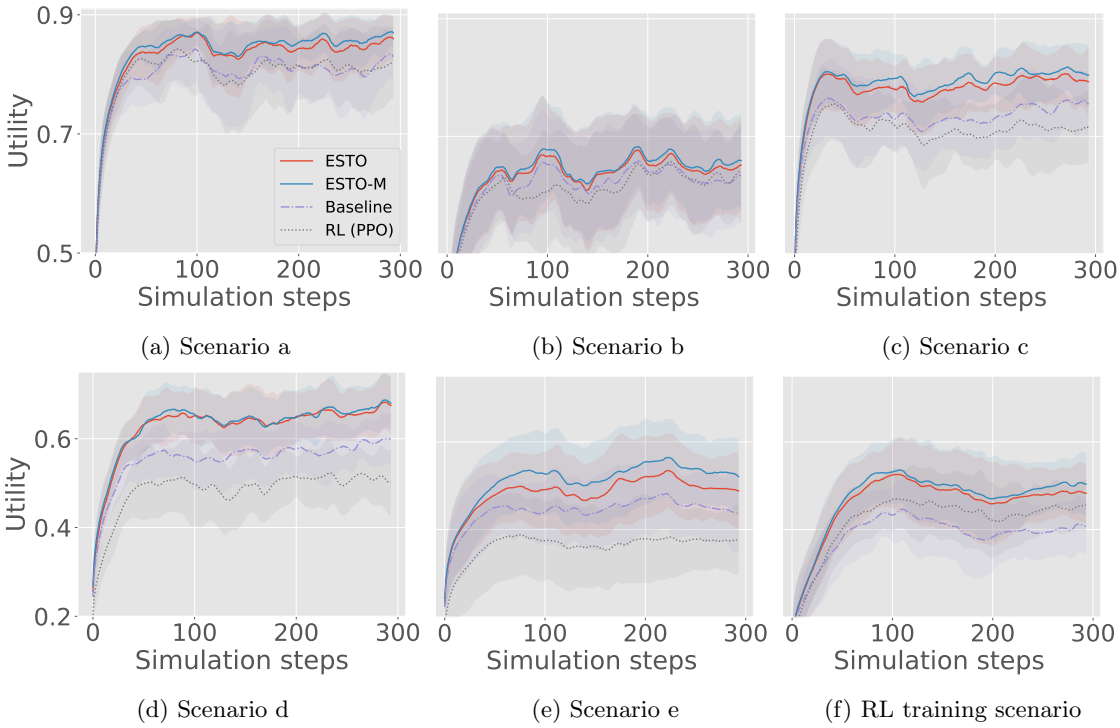
5 Evaluation

The evaluation is performed on a multi-agent simulator built with the mesa framework [16]. It consists of an environment of fixed size without obstacles with two kinds of agents: **the radars**



(a) Non saturated well positioned radars (b) Few saturated well positioned radars (c) Several saturated well positioned radars (d) Many saturated well positioned radars (e) Saturated ill-positioned radars

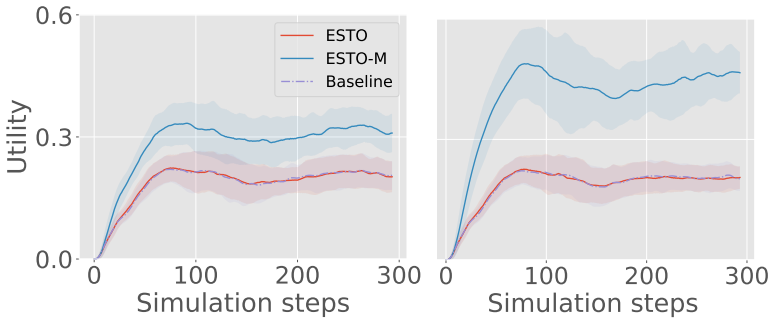
Figure 1: The 5 validation scenarios (radars & FOV in blue, targets in red)



(a) Scenario a (b) Scenario b (c) Scenario c

(d) Scenario d (e) Scenario e (f) RL training scenario

Figure 2: Utility of the radars over 300 steps with 16 random seeds



(a) 2 stacked radars (b) 8 stacked radars

Figure 3: Performance for stacked radars scenarios

are implemented according to the model presented in section 3. In order to simplify the model, we make the following assumptions: the radars rotate at the rate of 1 round/step; **the targets** have a fixed speed and direction. They can turn by up to 15° and are replaced when they leave the simulation space. The simulator uses a random scheduler *i.e.* the agents act in a random order. The information they use may therefore be outdated, which allows to check the system resilience when the agents don't have up-to-date information. The ESTO approach is optimized on a scenario with 3 fixed radars and 20 targets with random trajectories over 10 runs to enhance generalization. The RL agent trains on the same settings, but over one run only due to time constraints. Our approaches are compared to a simple "baseline" approach (the radars greedily select the closest targets) on the 5 scenarios provided in fig. 1, representing interesting configurations of the radar network.

As fig. 2 shows, both ESTO and ESTO-M significantly outperform the baseline on all scenarios. The performance gain seems to be correlated with the overlap of the agents field of view (FOV). When the FOV overlap is minimal, there is less need for cooperation between agents and the baseline is close to being optimal. Conversely, when the overlap is maximal, cooperation is needed to achieve a good performance. Indeed, when radars are stacked (fig. 3), ESTO-M performs significantly better than the baseline, even more so as more radars are stacked unlike ESTO. This indicates that the distance to the closest radar feature plays an important role in ESTO's collaboration. This is confirmed by the fact that we do not observe a significant difference between ESTO and ESTO-M in scenarios (a) to (e) when ESTO can use the feature. The RL approach relies on the reformulation of the problem (definition 1). It outperforms the baseline on the training scenario but seems to have poor generalization.

6 Conclusion

In this paper, we presented two novel reward-based learning algorithms for multi-radar multi-target allocation based on centralized learning and decentralized execution. The first one, ESTO, relies on CMA-ES to optimize a linear preference function that is used to order targets for a greedy selection. The second is an actor-critic based reinforcement learning approach relying on a specific Dec-POMDP formalization. While ESTO significantly outperforms our greedy baseline by learning cooperative behaviors, the RL approach still lacks generality to do so systematically. Training it longer on more diverse scenarios (target trajectory, radar positions, number of steps) may help to prevent overfitting. Moreover, future improvements may include: The development of a neural version of ESTO that would rely on a large scale CMA-ES implementation [17] to handle the increase in the size of the parameter space. Another improvement would be the development of a more realistic radar simulation taking into account *e.g.* changes in SNR and rotation speed, and include obstacles and targets of different classes and priorities. More importantly, other than simply tuning our models for better numerical performance, we would like to interface them with symbolic AI methods allowing them to leverage expert domain knowledge and opening the way for explainable AI (XAI) developments.

References

[1] Jinhui Dai, Junkun Yan, Shenghua Zhou, Penghui Wang, Bo Jiu, and Hongwei Liu. Sensor selection for multi-target tracking in phased array radar network under hostile environment. In *2020 IEEE Radar Conference (RadarConf20)*, 2020.

[2] Chenguang Shi, Lintao Ding, Wei Qiu, Fei Wang, and Jianjiang Zhou. Joint optimization of target assignment and resource allocation for multi-target tracking in phased array radar network. In *2020 IEEE Radar Conference (RadarConf20)*, 2020.

[3] Zhe Geng. Evolution of netted radar systems. *IEEE Access*, 8:124961–124977, 2020.

[4] Chenguang Shi, Fei Wang, Mathini Sellathurai, and Jianjiang Zhou. Non-cooperative game theoretic power allocation strategy for distributed multiple-radar architecture in a spectrum sharing environment. *IEEE Access*, 6:17787–17800, 2018.

[5] Haobo Jiang, Song Li, Chi Lin, Chuang Wang, Kaiqi Zhong, Guannan He, Qizhi Zhang, Yuanhao Zhao, and Jiayi Liu. Research on distributed target assignment based on dynamic allocation auction algorithm. In *Journal of Physics: Conference Series*, volume 1419, page 012001, 2019.

[6] Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning, 2017.

[7] Sriram Srinivasan, Marc Lanctot, Vinícius Flores Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 3426–3439, 2018.

[8] Pablo Hernandez-Leal, Bilal Kartal, and Matthew E. Taylor. A survey and critique of multiagent deep reinforcement learning. *Auton. Agents Multi Agent Syst.*, 33(6):750–797, 2019.

[9] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments, 2017.

[10] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula, 2019.

[11] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning, 2016.

[12] Nikolaus Hansen and Andreas Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evolutionary computation*, 9(2):159–195, 2001.

[13] Frans Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. Springer, 01 2016.

[14] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2137–2145, 2016.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[16] David Masad and Jacqueline Kazil. Mesa: an agent-based modeling framework. In *14th PYTHON in Science Conference*, pages 53–60, 2015.

[17] Konstantinos Varelas, Anne Auger, Dimo Brockhoff, Nikolaus Hansen, Ouassim Ait ElHara, Yann Semet, Rami Kassab, and Frédéric Barbaresco. A comparative study of large-scale variants of cma-es. In *International Conference on Parallel Problem Solving from Nature*, pages 3–15, 2018.