Aircraft Trajectory Planning for Climate Hotspot Avoidance Considering Air Traffic Complexity: A Constrained Multi-Agent Reinforcement Learning Approach

Fateme Baneshi, María Cerezo Magaña, Manuel Soler
Department of Aerospace Engineering
Universidad Carlos III de Madrid
Madrid, Spain

Tingting Ni, Maryam Kamgarpour
Automatic Control Laboratory
EPFL
Lausanne, Switzerland

***Abstract*—Planning aircraft trajectories to avoid climate-sensitive areas poses operational challenges, including increased traffic complexity and potential safety risks. This study presents a framework designed to plan operationally feasible climate-friendly routes from the perspective of the air traffic management (ATM) system. The problem is formulated as a constrained Markov game, where air traffic complexity, a key indicator of air traffic manageability, serves as the objective function, and climate hotspot avoidance is imposed as a constraint. The proposed method employs the multi-agent proximal policy optimization algorithm and adapts it to handle constraints related to climate hotspot avoidance using the Lagrangian technique. To ensure scalability, parameter sharing is employed, allowing the algorithm to deal with varying numbers of concurrently operating aircraft in different scenarios. Experimental results demonstrate that the proposed algorithm effectively balances environmental goals with traffic manageability, offering operationally feasible climate-optimal trajectories.**

***Keywords*—Climate impact, Aircraft trajectory optimization, Air traffic management system, Multi-agent reinforcement learning, Constrained Markov decision process, Proximal policy optimization algorithm.**

I. Introduction

The aviation sector is a major contributor to climate change through CO_2 emissions and various non-CO_2 forcing agents [1]. Recent studies highlight that non-CO_2 emissions from aviation account for approximately two-thirds of the sector's total contribution to global warming [1], [2]. Key contributors include the emissions of nitrogen oxides and water vapor and the formation of persistent contrails. The climate impact of these non-CO_2 species highly depends on meteorological conditions at the time and location of emissions, making flight planning a promising short/medium-term strategy for mitigating their corresponding effects [3].

Extensive research has explored leveraging these spatiotemporal dependencies to plan aircraft trajectories in a more climate-friendly manner [4], [5]. Although these approaches show significant potential for reducing climate impact, they primarily focus on optimizing individual flight trajectories (see [6]). This narrow focus overlooks complex interactions between flights and the overall manageability of air traffic, raising concerns about the practical feasibility of such optimized flight plans [7].

As highlighted in [7], optimizing individual flight trajectories without considering their interactions can negatively impact air traffic manageability. To better understand the climate impact mitigation potential achievable through flight planning, it is essential to conduct analyses at the network scale, considering the collective behavior of all flights within the system. In [8], we made the first attempt to address this challenge in the literature by proposing a two-stage method. Initially, we optimized each individual trajectory in a climate-friendly manner (micro-level flight planning). Then, we minimized potential conflicts by making slight adjustments to the speed profile of the optimized routes [8]. However, this approach has drawbacks, particularly in terms of computational cost. Optimizing aircraft trajectories at the micro-level is computationally very intensive [6]. Additionally, the subsequent phase, which modifies trajectories to maintain ATM performance, adds additional computational complexity. Moreover, adjustments made to flight profiles during the second phase may unnecessarily compromise the optimal performance achieved in the initial trajectory optimization due to the absence of a feedback scheme between the two stages.

In this work, we propose a single-step optimization algorithm that simultaneously addresses both climate impact mitigation and air traffic manageability. In the proposed algorithm, each aircraft has a dual objective: reducing climate impact and maintaining air traffic manageability. To mitigate climate impact, we first identify specific airspace regions where aircraft emissions have significant warming effects, termed 'ECHO' areas. These regions are then incorporated as constraints that aircraft should avoid [3]. To ensure the operational feasibility of trajectories, traffic complexity is considered as the objective function to be minimized. Starting from business-as-usual (BAU) trajectories, each aircraft adjusts

its flight path to avoid ECHO areas while minimizing the overall air traffic complexity.

Solving this multi-agent control problem presents significant complexities due to the large number of agents involved (thousands of aircraft), each with multiple state and control variables [4]. Each aircraft's trajectory follows a highly nonlinear 3D point mass model, representing the aircraft's dynamical behavior, making the optimization problem difficult to solve directly. Additionally, the objective of minimizing air traffic complexity creates inter-agent dependencies; mathematically, this coupling requires incorporating additional terms in each agent's objective function to account for the states of neighboring aircraft. This dependency transforms the problem into a large-scale, coupled optimization, where each agent's optimal path cannot be computed separately. Furthermore, each scenario often requires a separate solution, as control inputs must be optimized for specific conditions and constraints.

Multi-agent reinforcement learning (MARL) simplifies the multi-agent control problem by utilizing policy learning, where agents learn policies through training. MARL reduces high-dimensional state-action spaces by using function approximation, like neural networks, to generalize from sampled experiences, enabling agents to make decisions across a broad state space without explicitly exploring each possible combination [9]. The MARL framework can inherently manage complex dynamics by training agents through interactions with the environment in a model-free manner, eliminating the need to accurately model system dynamics [10]. By designing rewards that account for air traffic complexity, MARL implicitly fosters inter-agent coordination without the need for explicit optimization of coupled objectives [9]. Furthermore, for new scenarios, agents can leverage the trained policy for decision-making without the need to solve the problem from scratch.

To efficiently address the climate-optimal flight planning problem at the ATM network scale, we introduce a framework based on constrained multi-agent reinforcement learning (MARL). Each aircraft is treated as an agent, and the entire airspace, encompassing all aircraft, is modeled as an environment with multiple decision-makers. While extensive research has been conducted on unconstrained MARL [10], [11], constrained MARL remains less explored. The authors in [12] proposed a constrained MARL approach; however, it faces several limitations. Applying such methods to large-scale domains, such as aviation with thousands of agents, requires training numerous networks, which becomes computationally expensive. Additionally, the turn-based action selection process in this approach may be inefficient in multi-agent environments, as agents must wait for others to act, resulting in delays and coordination challenges. Moreover, the study in [12] does not account for environments with partial observability, restricting its applicability in real-world scenarios.

In this study, we present a cooperative framework that utilizes multi-agent proximal policy optimization (MAPPO) [13]. MAPPO has demonstrated superior performance in various cooperative multi-agent games [13] and has been successfully applied to diverse areas, such as unmanned aerial vehicles [14] and air traffic control systems [15]. Building on its success, we adapt MAPPO to incorporate constraint handling for individual agents through the Lagrangian approach. This adaptation allows us to effectively balance the dual objectives of minimizing climate impact and ensuring air traffic manageability. We employ a centralized learning and decentralized execution scheme, which enables efficient coordination among multiple aircraft [16]. Recognizing the need for scalability in scenarios involving varying numbers of agents, we implement shared policy parameters, ensuring flexibility across diverse air traffic situations. Our contributions are summarized as follows:

- Introducing an optimization framework to plan climate-optimized trajectories while simultaneously ensuring their feasibility from the ATM perspective.
- Proposing a constrained MARL framework that employs the MAPPO algorithm and adapts it to handle constraints related to climate hotspot avoidance.

We evaluate the proposed framework through experiments using real traffic data within European airspace. We compare the performance of the constrained MAPPO algorithm against two MAPPO variants, each optimized either for complexity reduction or for reducing climate-hotspot violations. The results highlight that optimizing for one objective (i.e., complexity reduction or hotspot avoidance) does not necessarily address the other. In contrast, the proposed constrained MAPPO successfully avoids a large number of climate hotspots while reducing complexity to levels even lower than those of business-as-usual trajectories, demonstrating its potential to plan feasible, climate-optimal paths in a computationally efficient manner.

II. Constrained Multi-Agent Reinforcement Learning

A. Partially observable constrained Markov decision process

We formulate the problem as a partially observable constrained Markov decision process (POCMDP) defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \gamma, R, \{C^i\}_{i=1}^N, \{c^i\}_{i=1}^N, s_0 \rangle$. Here, $\mathcal{N} = \{1, \dots, N\}$ represents the set of agents, $\mathcal{S}$ is the combined state space for all agents, $\mathcal{A}^i$ denotes the action space for agent i, $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ is the joint action space for all agents, $o^i = \mathcal{O}(\mathcal{S}, i)$ represents the local observation for agent i at state s, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the state transition probability function, $\gamma \in [0, 1)$ is the discount factor, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the shared reward function, $C^i : \mathcal{S} \times \mathcal{A}^i \to \mathbb{R}$ is the cost function for agent i, with a cost threshold c^i. In this fully cooperative setting, the reward function R depends on the joint actions of all agents, reflecting its coupling across agents. The constraints are decoupled, as each agent's cost C^i depends only on its own actions a^i.

We denote the initial state of all agents combined by $s_0 \in \mathcal{S}$. At each time step t, agent i observes o_t^i and selects an action a_t^i according to a randomized stationary policy $\pi^i(\cdot|o_t^i) \in \Pi^i$. The joint action $\mathbf{a}_t = (a_t^1, \dots, a_t^N)$ is then executed, transitioning the system to a new state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, \mathbf{a}_t)$. Each agent i then receives a reward $R(s_t, \mathbf{a}_t)$ and incurs a cost $C^i(s_t, a_t^i)$.

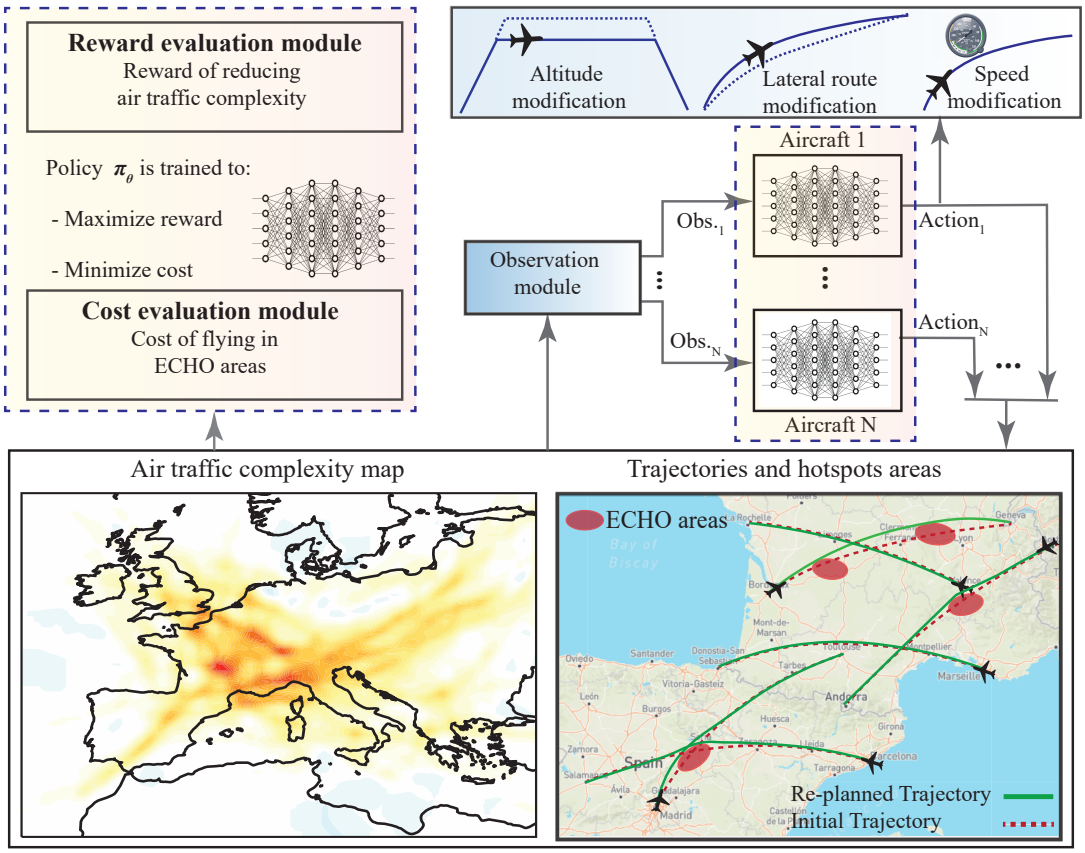


Figure 1. Overview of the proposed framework. Each aircraft receives a local observation about the surrounding traffic and the location of climate-sensitive hotspots (ECHO areas). Based on these observations, the aircraft executes actions according to a trained policy aimed at avoiding hotspots while maintaining manageable traffic complexity. Note: lateral route modifications are illustrated here to represent the full scope of potential adjustments; however, only vertical and speed changes were implemented in this study.

The set of joint policies is denoted by $\pi = \{\pi_i\}_{i\in\mathcal{N}}$ and is represented as $\Pi := \Pi^1 \times \cdots \times \Pi^N$. For any joint policy $\pi \in \Pi$, we define the reward value function at state s as $V_R^\pi(s) := \mathbb{E}_{\mathbf{a}_t\sim\pi, s_t\sim\mathcal{P}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,\mathbf{a}_t) \mid s_0 = s\right]$ and cost value function at state s as $V_{C^i}^\pi(s) := \mathbb{E}_{\mathbf{a}_t\sim\pi, s_t\sim\mathcal{P}}\left[\sum_{t=0}^{\infty}\gamma^t C^i(s_t,a_t^i) \mid s_0 = s\right]$ for $i \in \mathcal{N}$.

The goal is to find a policy that maximizes the reward value function $V_R^\pi(s_0)$ while ensuring that the constraint $V_{C^i}^\pi(s_0) \leq c^i$ is satisfied for every agent i. Formally, this is expressed as:

$$\max_{\pi\in\Pi} V_R^\pi(s_0), \text{ s.t. } V_{C^i}^\pi(s_0) \leq c^i, \forall i \in \mathcal{N}. \tag{1}$$

In this study, we focus on a large population of agents that are assumed to be homogeneous. This assumption is justified by a common reward function that aligns all agents' interests toward minimizing air traffic complexity while avoiding climate-sensitive regions. Such homogeneity also implies that the agents play interchangeable roles in the system's evolution and are nearly indistinguishable from each other [17]. Due to the homogeneity of the agents, parameter sharing can be applied to enhance scalability and training efficiency [18]. This allows all agents to use a single shared policy [13], [19]. The shared policy, denoted by π_θ and parameterized by θ, enables training to utilize the collective experience of all agents. Meanwhile, each agent i can still take its own actions, $\pi_\theta(\cdot|o_t^i)$, based on its observations o_t^i [20].

B. Constrained multi-agent proximal policy optimization

Solving constrained MARL problems involves several challenges. These include non-stationarity, where the environment evolves dynamically in response to the actions of multiple agents: scalability, where computational complexity grows exponentially with the number of agents; and training stability, where large policy updates can lead to instability and performance collapse by erasing previously learned good behaviors. Additionally, balancing reward optimization with constraint satisfaction further complicates the problem.

To tackle these challenges in an unconstrained setting, [13] introduced the MAPPO algorithm, which extends the proximal policy optimization (PPO) [21] framework from single-agent to multi-agent environments. MAPPO employs separate neural networks for the policy π_θ and the value function $V_\phi(s)$. The value function helps to reduce the variance during training. MAPPO follows a centralized training with decentralized execution approach and has shown superior performance in various multi-agent scenarios, such as Google Research Football, the StarCraft Multi-Agent Challenge, and Hanabi [13].

In this study, we extend the MAPPO to address constrained optimization. For single-agent reinforcement learning, [22] utilized the Lagrangian technique to incorporate constraints directly into the PPO objective function, achieving reliable safety performance. Inspired by this approach, we combine the Lagrangian method with MAPPO to address the constrained multi-agent problem defined in 1. We formulate as the following min-max problem:

$$\max_{\theta} \min_{\lambda^i \geq 0,\, i \in \mathcal{N}} V_R^{\pi_\theta}(s_0) - \sum_{i \in \mathcal{N}} \lambda^i (V_{C^i}^{\pi_\theta}(s_0) - c^i). \quad (2)$$

To solve this min-max problem, we apply gradient descent on the Lagrange multipliers $\{\lambda^i\}_{i\in\mathcal{N}}$ and gradient ascent on the policy parameters θ. However, directly applying gradient ascent on θ can lead to large, unstable updates, potentially causing the policy to forget previously learned good behaviors, which results in performance collapse. The PPO framework mitigates this issue by employing trust region optimization, which constrains the magnitude of policy updates. PPO achieves this by clipping the probability ratio $\frac{\pi_\theta}{\pi_{\theta_{\text{old}}}}$ within $(1-\epsilon, 1+\epsilon)$, ensuring that the new policy π_θ remains close to the old policy $\pi_{\theta_{\text{old}}}$. This clipping mechanism enhances stability and enables more controlled updates.

To introduce the MAPPO method [13], we define the reward state-action value function $Q_R^{\pi_\theta}(s, \mathbf{a}) := \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta, s_t \sim \mathcal{P}}[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t) | s_0 = s, \mathbf{a}_0 = \mathbf{a}]$ and the cost state-action value function $Q_{C^i}^{\pi_\theta}(s, a^i) := \mathbb{E}_{\mathbf{a}_t \sim \pi_\theta, s_t \sim \mathcal{P}}[\sum_{t=0}^{\infty} \gamma^t C^i(s_t, a_t^i) | s_0 = s, a_0^i = a^i]$ for $i \in \mathcal{N}$. And the advantage function is defined as $A_u^{\pi_\theta}(s,a) := Q_u^{\pi_\theta}(s,a) - V_u^{\pi_\theta}(s)$ for $u \in \{R\} \cup \{C^i | i \in \mathcal{N}\}$. This function evaluates the benefit of taking action a in state s relative to the baseline value $V_u^{\pi_\theta}(s)$. Using these definitions, the MAPPO objective is formulated as:

$$L\left(\theta, \{\lambda^i\}_{i\in\mathcal{N}}\right) := \mathbb{E}_{\mathbf{a}\sim\pi_\theta, s\sim p}\Bigg[\sum_{i=1}^{N} \min\left(\frac{\pi_\theta(a^i \mid o^i)}{\pi_{\theta_{\text{old}}}(a^i \mid o^i)} A_{\lambda^i}^{\pi_\theta}(s, \mathbf{a}),\right.$$
$$\left.\text{clip}\left(\frac{\pi_\theta(a^i \mid o^i)}{\pi_{\theta_{\text{old}}}(a^i \mid o^i)}, 1-\epsilon, 1+\epsilon\right) A_{\lambda^i}^{\pi_\theta}(s, \mathbf{a})\right)\Bigg], \quad (3)$$

where $A_{\lambda^i}^{\pi_\theta}(s, \mathbf{a}) := \frac{A_R^{\pi_\theta}(s,\mathbf{a})}{N} - \lambda^i \left(A_{C^i}^{\pi_\theta}(s, a^i) - c^i\right)$, and the advantage functions are computed using generalized advantage estimation . Here, λ^i penalizes constraint violations.

To solve problem (2), we iteratively apply the following update rules:

$$\lambda^i \leftarrow \lambda^i - \alpha_\lambda \nabla_{\lambda^i} L\left(\theta, \{\lambda^i\}_{i\in\mathcal{N}}\right), \forall i \in \mathcal{N},$$
$$\theta \leftarrow \theta + \alpha_\theta \nabla_\theta L\left(\theta, \{\lambda^i\}_{i\in\mathcal{N}}\right),$$

where α_λ and α_θ are the learning rates for updating $\{\lambda^i\}_{i\in\mathcal{N}}$ and θ, respectively. These updates balance constraint satisfaction with reward maximization at the individual agent level.

III. Casting climate optimal trajectory planning at network scale as a constrained MARL problem

In this section, we outline the key components of the MARL framework used to solve the flight planning problem for the benefit of climate. Specifically, we define the observation space, action space, reward function, and cost function.

A. Observation space

In our multi-agent reinforcement learning framework, we model the environment as a POCMDP. The state s_t at time t encapsulates all information about the environment, including the positions, velocities, headings, and trajectories of all aircraft, as well as the locations and characteristics of climate hotspots.

Each aircraft i receives a local observation o_t^i, which is a function of the state s_t. The local observation o_t^i for each aircraft comprises the following components:

- Trajectory information τ_t^i: Detailed data on the aircraft's discretized trajectory:
$$\tau_t^i = \left[(\varphi_0^i, \lambda_0^i, h_0^i)_t, \ldots, (\varphi_k^i, \lambda_k^i, h_k^i)_t\right],$$
where the tuple $(\varphi_l^i, \lambda_l^i, h_l^i)_t$ represents the latitude [hft], longitude [degree], and altitude [degree] at each discretized point l and time step t.
- Flight parameters: Information including heading angle χ_t^i, flight phase p_t^i, speed v_t^i [m/s], and the duration T_t^i over which the aircraft i flies the most complex grid segment at time step t.
- Information about neighboring aircraft I_t^i: Relative information about neighboring aircraft within a certain vicinity, which aids in assessing local air traffic complexity. We denote $(r_v^l, r_\chi^l, r_p^l)_t$ as the relative speed, heading difference, and phase difference with respect to the agent i for each neighboring aircraft l at time step t. The information about neighboring aircraft for agent i is given by:
$$I_t^i = \left((r_v^1, r_\chi^1, r_p^1)_t, \ldots, (r_v^m, r_\chi^m, r_p^m)_t\right),$$
where m is the number of neighboring aircraft.
- Climate hotspot information E: Coordinates of the centers of climate hotspot areas, which are common observations available to all agents:
$$E = \left[(\varphi_{e_1}, \lambda_{e_1}, h_{e_1}), \ldots, (\varphi_{e_{n_h}}, \lambda_{e_{n_h}}, h_{e_{n_h}})\right],$$
where n_h is the number of hotspots, and $(\varphi_{e_j}, \lambda_{e_j}, h_{e_j})$ represents the location of hotspot j.

Based on the above, the local observation for agent i at time step t is defined as:

$$o_t^i = \left[\tau_t^i, \chi_t^i, p_t^i, v_t^i, T_t^i, I_t^i, E\right].$$

B. Action space

In this study, the action space for each agent is defined by the potential modifications to the aircraft's trajectory. Specifically, agents can execute predefined maneuvers to meet defined objectives. These maneuvers are categorized into two primary types: speed adjustments and altitude changes. Agents can alter their speed by increasing or decreasing the Mach number by

0.03 or by maintaining their current velocity. Similarly, altitude adjustments involve increasing or decreasing the flight level by 20 hft, or maintaining the current altitude. The selected action is applied uniformly to all points of the trajectory. For instance, if an agent chooses to increase its altitude, the entire flight profile τ^i is updated to reflect the new altitude. Accordingly, all information related to the trajectory, such as information on neighboring aircraft, is updated.

C. Reward function

The reward function R provides the immediate reward received by all agents for transitioning from state s to state s' due to the joint action $\mathbf{a}$. In this study, the reward function is defined by the traffic complexity score. The complexity score serves as an indicator of the difficulty and effort required to effectively monitor and manage air traffic situations. It is calculated based on three key metrics: vertical (ν), horizontal ($\varkappa$), and speed (υ) differences interacting flows. ν captures vertical maneuvers and represents the complexity of managing flights with varying flight phases (managing mixed-phase traffic is more challenging than handling aircraft in similar phases (e.g., only cruising)). $\varkappa$ reflects the complexity of handling intersecting flows, which is inherently more complex than managing parallel flows. Finally, υ represents speed variations among aircraft, with the assumption that similar speeds correlate with lower complexity. These indicators collectively provide insights into potential hazards within the airspace, focusing on the duration and severity of interactions rather than solely the presence of aircraft in the same volume.

To assess traffic complexity, the airspace is divided into identical 4D grids, representing time and 3D spatial dimensions. Two aircraft are considered to be interacting at any given time if they are located within the same cell from each aircraft's perspective [23]. The complexity for each aircraft i with respect to aircraft k is computed as follows:

$$\Psi_t^{i,k} = \sum_{g_t}^{g_{t+\Delta t}} (\nu^{i,k} + \varkappa^{i,k} + \upsilon^{i,k})$$

where g_t is the grid that aircraft i enters at time t, and $g_{t+\Delta t}$ is the cell that aircraft i exits at time $t+\Delta t$. The sum operator encompasses all cells that aircraft i crosses between g_t and $g_{t+\Delta t}$. The variables $\nu^{i,k}$, $\varkappa^{i,k}$ and $\upsilon^{i,k}$ are computed as:

$$\nu^{i,k} = \begin{cases} \frac{2\kappa^2}{(t_x{}^i - t_e{}^i)+(t_x{}^k - t_e{}^k)} & \text{if } \kappa \neq \varnothing \text{ and } P^i \neq P^k \\ 0 & \text{otherwise} \end{cases}$$

$$\varkappa^{i,k} = \begin{cases} \frac{2\kappa^2}{(t_x{}^i - t_e{}^i)+(t_x{}^k - t_e{}^k)} & \text{if } \kappa \neq \varnothing \text{ and } |\chi^i - \chi^k| > 20^\circ \\ 0 & \text{otherwise} \end{cases}$$

$$\upsilon^{i,k} = \begin{cases} \frac{2\kappa^2}{(t_x{}^i - t_e{}^i)+(t_x{}^k - t_e{}^k)} & \text{if } \kappa \neq \varnothing \text{ and } |v^i - v^k| > 35 \text{ kts} \\ 0 & \text{otherwise} \end{cases}$$

where t_e and t_x are the entering and exit times of aircraft within the cell, respectively. P represents the flight phase, χ denotes the heading angle, v is the true airspeed, and κ represents the time overlap between the two aircraft, which is defined as: $\kappa = [t_e^i, t_x^i] \cap [t_e^k, t_x^k]$. The overall reward function is then calculated as follows:

$$R_t = \Psi_0 - \sum_{i=1}^{N} \sum_{k=1, k\neq i}^{N} \Psi_t^{i,k}$$

where Ψ_0 is the initial complexity score.

D. Constraints

The non-CO_2 climate impact of aircraft emissions, including contrail formation, nitrogen oxides-induced changes in atmospheric concentrations of methane and ozone, and water vapor emissions, exhibits significant spatiotemporal variability. Therefore, we can mitigate their corresponding climate effects by planning aircraft trajectories to avoid areas where the emissions have a large climate impact. These areas, often referred to as climate hotspots or ECHO areas, are generally regions where the net non-CO_2 climate effects exceed predefined thresholds. Interested readers are referred to [3] for an approach to identify such areas.

In this study, to achieve trajectories that are climatically friendly, we model the avoidance of climate hotspots as constraints C^i to be satisfied. We define the following performance metric to quantify the cost of constraint violation (i.e., the intersection of flight trajectories with the climate sensitive areas) to be included in Eq. (3):

$$C_t^i = \begin{cases} c_h & \text{if } \tau_t^i \in E \\ 0 & \text{otherwise} \end{cases}.$$

Here, τ_t^i represents the trajectory of aircraft i during time interval $[t, t+\Delta t]$, c_h is a constant cost, and E denotes ECHO areas. The equation implies that when an aircraft flies through climate-sensitive areas, a cost c_h is incurred.

IV. Case Study

We performed an experiment utilizing a real traffic scenario over ECAC[1] airspace on December 20, 2018. The case study includes all the flights within ECAC airspace from 12:00 UTC to 16:00 UTC. The weather data, including wind and temperature, was obtained from the ERA5 reanalysis data products available at the Copernicus Data Store[2]. The initial flight trajectories were obtained using our in-house tool, ROOST. [3] Although the initial trajectories in this study were generated using ROOST, the framework is flexible and can use other planned trajectories, such as those available from Eurocontrol's demand data repository (DDR2) dataset[4]. Each aircraft's trajectory includes detailed information on latitude, longitude, altitude, time, true airspeed, Mach number, mass, heading angle, and flight phase.

[1]European Civil Aviation Conference

[2]https://cds.climate.copernicus.eu/cdsapp\#!/dataset/reanalysis-era5-pressure-levels?tab=form

[3]ROOST is publicly available and accessible via DOI: https://doi.org/10.5281/zenodo.7495472

[4]https://www.eurocontrol.int/ddr

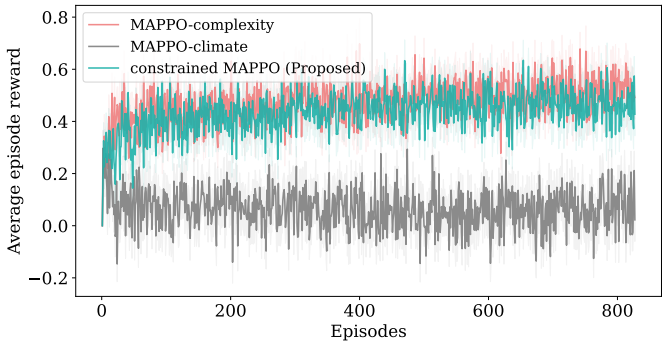


Figure 2. Performance comparisons in terms of the reward.

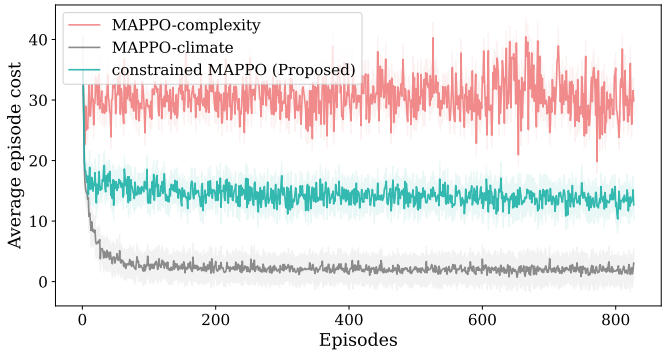


Figure 3. Performance comparison in terms of the cost. We set the cost threshold $c^i = 0$ for all $i \in \mathcal{N}$.

Given the computational complexity of the traffic scenario, which involves approximately 6,000 flights, we adopted a strategy to reduce the computational burden by generating random subsets of the data. Specifically, we selected random portions of airspace measuring 500 by 500 nautical miles within a one-hour time frame. All flights intersecting these regions during the specified period were grouped into subsets. Each subset contains 90–120 flights. This approach allows a flight to appear in multiple groups, each with different group members, ensuring coverage of the entire airspace and providing sufficient variability in the training data.

As an initial step and proof of concept, this study employs fictitious climate hotspots in the airspace; however, the proposed approach can seamlessly accommodate real hotspots without loss of generality. These hotspots are generated across flight levels ranging from FL280 to FL450, with a 75% probability of occurring. Their locations are randomly assigned within the airspace, and they are modeled as ellipsoids with a radius of 20 nautical miles and a vertical extent of 1000 ft. For simplicity, a uniform cost is applied whenever an aircraft enters a hotspot, regardless of the distance flown within it. Future work will extend the model to incorporate real hotspots and account for the distance aircraft travel within these regions. Nevertheless, real hotspots can also be modeled as ellipsoids.

Once all preprocessing is finalized, including trajectory clustering and hotspot generation, the proposed strategy outlined

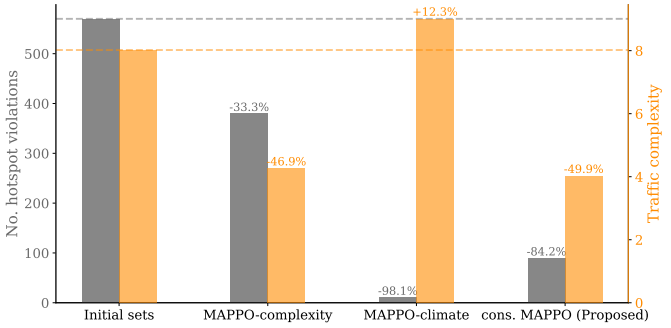


Figure 4. Performance comparison of trained policies across three algorithms over 100 test sets, each containing 90–120 flights.

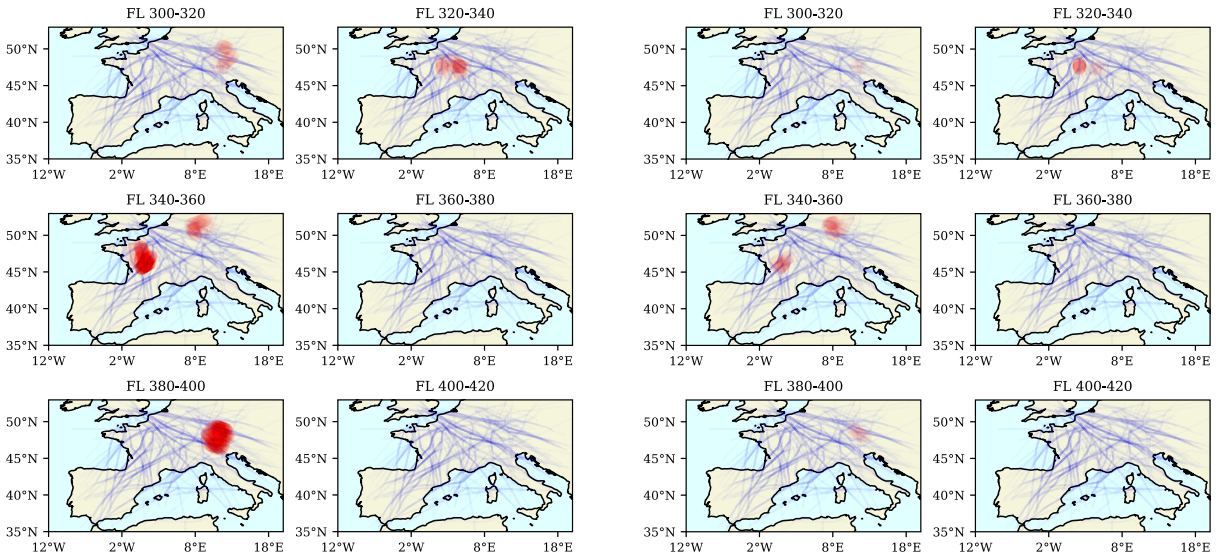
TABLE I. HYPERPARAMETERS USED IN THE EXPERIMENT.

Hyperparameters	Values	Hyperparameters	Values
Critic lr	1e-3	Batch size	124
Actor lr	1e-3	N mini-batch	32
Safety_bound	0	ϵ	0.2
Eval. episodes	100	Lagrangian coef. rate	0.1
Optimiser	Adam	c_h	10

in Section II-B is implemented. The state space, action space, reward function, and cost function, as described in Section III, are incorporated into our experimental framework. In this study, the time step is set to 60 minutes.

We initialize the policy network π_θ. The critic and cost-critic networks are set up to compute the advantage and cost-advantage functions in Eq. (3), respectively. All networks (i.e., policy network, critic, and cost-critic) share a similar architecture, each designed as a three-layer multi-layer perceptron (MLP) with two hidden layers of 264 units each. The actor network's final layer produces action probabilities via a softmax activation function, whereas the final layers of the critic and cost-critic networks output single scalar values representing the estimated cumulative reward and cost, respectively. ReLU activation functions are employed between layers, and orthogonal initialization is used for all networks to enhance training stability. Detailed of the hyperparameters used in the experiment is presented in Table I.

The proposed 'constrained MAPPO' algorithm is compared against two variations of the MAPPO algorithm, each targeting different optimization objectives. The first variation, 'MAPPO-complexity,' focuses only on optimizing traffic complexity, while the second, 'MAPPO-climate', considers avoiding climate hotspots. Figure 2 presents a comparison of reward performance for the three algorithms. The vertical axis represents the sum of the rewards for all agents within a set, and the reward curve is smoothed over 300 episodes for better clarity. As observed in Fig. 2, the MAPPO-complexity algorithm consistently achieves a higher reward, as it focuses solely on minimizing traffic complexity. In contrast, the MAPPO-climate algorithm, which only considers climate impact by avoiding hotspots, exhibits low reward performance due to the lack of emphasis on traffic complexity. The proposed constrained MAPPO algorithm achieves a competitive reward performance relative to MAPPO-complexity.



(a) Initial trajectories

(b) Re-planned trajectories using Constrained MAPPO

Figure 5. Hotspot violations for both the initial and re-planned trajectories across various flight levels. The red areas represent regions where aircraft trajectories intersect with climate-sensitive hotspots.

Figure 3 presents a comparison of the episode cost of violation for the three algorithms: MAPPO-complexity, MAPPO-climate, and the proposed constrained MAPPO. The vertical axis shows the average episode cost for all agents, with the safety constraint value set to 0. As seen in the figure, the MAPPO-complexity algorithm, which does not prioritize climate constraints, incurs the highest cost due to significant violations of climate hotspots. On the other hand, the MAPPO-climate algorithm, which focuses solely on minimizing climate impact, maintains very low costs, as it effectively avoids climate hotspots. The proposed constrained MAPPO algorithm achieves a balance between these two objectives. While its cost is higher than MAPPO-climate, it still maintains climate hotspot avoidance at reasonable levels, while optimizing for traffic complexity. This trade-off demonstrates the effectiveness of constrained MAPPO in managing the conflicting objectives of minimizing both the cost of hotspot violation and traffic complexity.

The policies derived from the three algorithms were evaluated on 100 test sets to assess their performance in balancing traffic complexity and hotspot avoidance. As depicted in Fig. 4, the proposed constrained MAPPO algorithm successfully reduces both hotspot violations and traffic complexity, demonstrating its capability to manage these conflicting objectives simultaneously. In contrast, the MAPPO-complexity algorithm, which focuses solely on optimizing traffic complexity, improves traffic manageability but results in a high number of hotspot violations. On the other hand, the MAPPO-climate algorithm, which only considers hotspot avoidance, effectively minimizes hotspot violations but at the expense of increased traffic complexity.

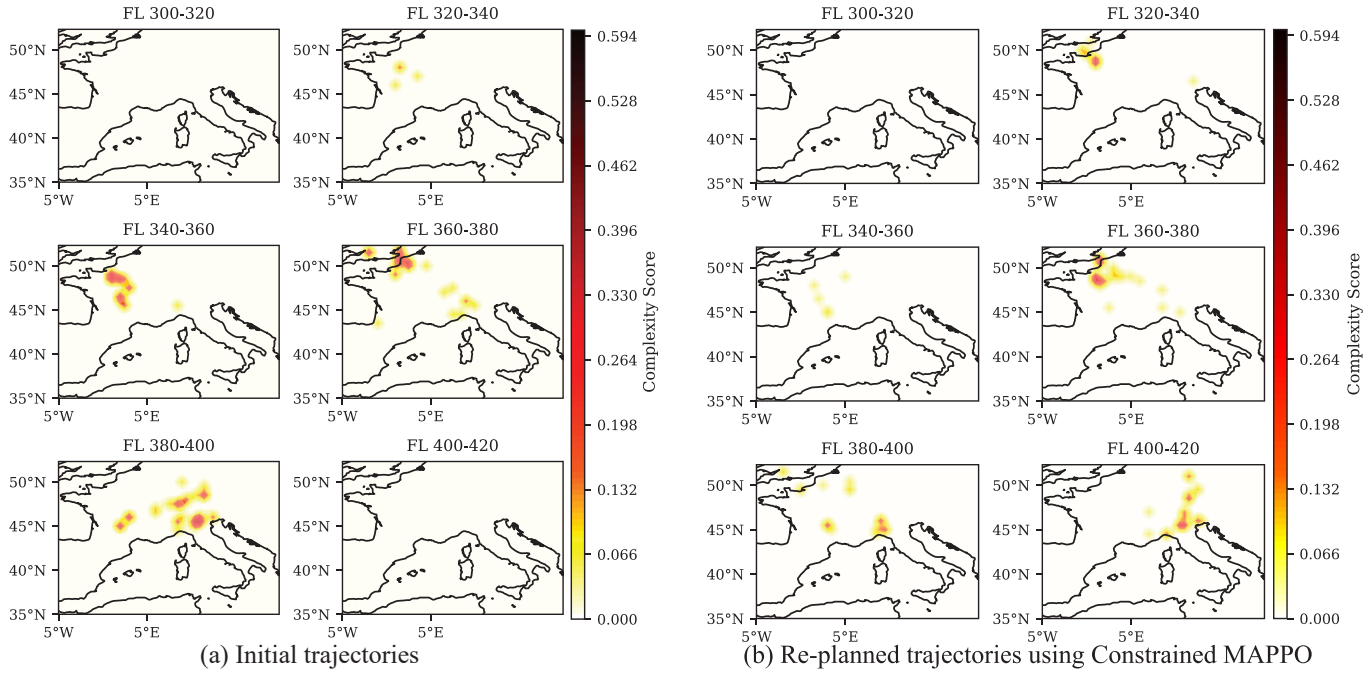
To provide a visual representation of the proposed approach's performance, a subset of the traffic within the latitude range [37,52] and longitude range [-5,15] between 13:00 UTC and 14:00 UTC was selected. The hotspot violations of the initial trajectories and those re-planned using the constrained MAPPO algorithm are depicted in Fig. 5, where the red areas indicate parts of the trajectories that crossed hotspots. Additionally, the traffic complexity for this set is presented in Fig. 6. The results indicate that complexity can be reduced to levels even lower than those of business-as-usual trajectories, showing the algorithm's potential for planning feasible, climate-optimal routes.

V. Discussion

The results underscore a trade-off between the two objectives: reducing traffic complexity and minimizing hotspot violations. The challenge, therefore, lies in finding an optimal balance that addresses both concerns simultaneously. The proposed constrained MAPPO algorithm successfully manages this trade-off.

However, there are some limitations in this study that highlight opportunities for future enhancement. In this study, airspace users (AU) preferences, such as cost efficiency and punctuality, were not directly incorporated (though the initial trajectories are cost-optimal trajectories). Yet, the proposed framework is capable of handling multiple constraints. Future research could introduce additional operational constraints, such as limiting increases in fuel consumption and flight time within specified thresholds. This would allow us to address AU preferences for operational cost control as an added constraint, similar to hotspot avoidance, thereby enhancing the model's practical applicability.

We employed parameter sharing for homogeneous agents primarily to enhance scalability and adaptability and reduce the number of training parameters, which was essential given



(a) Initial trajectories (b) Re-planned trajectories using Constrained MAPPO

Figure 6. Comparison of traffic complexity associated with the initial trajectories (a) and the re-planned trajectories using constrained MAPPO (b). The color scale indicates the complexity score, with darker regions representing areas of higher traffic complexity.

the large-scale nature of the problem. However, the framework could be extended to accommodate heterogeneous agents, where each aircraft would have its own unique parameters rather than sharing them across agents. While this would allow for more individualized optimization, it would also increase memory requirements for storing each agent's policy and could limit the model's adaptability to different scenarios.

One of the limitations of this study is the use of fictitious hotspots rather than real ones. However, real hotspots could seamlessly be incorporated into this framework. Looking ahead, future research will focus on extending the model to incorporate state-of-the-art climate impact estimation models (e.g., algorithmic climate change functions and Contrail Cirrus prediction (CoCiP) model) to determine actual climate hotspots and allow for their dynamic changes due to varying meteorological conditions. However, scenarios with high hotspot density would likely reduce flexibility in managing traffic complexity, as avoiding multiple hotspots may constrain trajectory options. Additionally, we aim to enhance the algorithm by integrating more decision variables, such as lateral paths, to further improve the flexibility and efficiency of the proposed methodology, making it more adaptable to complex, real-world scenarios.

VI. Conclusion

This paper introduced a novel approach for mitigating the environmental impact of aviation at the network scale by employing constrained multi-agent reinforcement learning. Our findings demonstrate that constrained MARL is a viable and efficient strategy for achieving more sustainable aviation operations. Conventional two-step optimization methods are time-consuming and risk losing the initial optimality when modifying flight plans to maintain air traffic manageability. The proposed approach simplifies this process by embedding constraints directly within the optimization framework, enabling the optimization of pre-planned trajectories in a single step. Simulation results showed that the proposed approach could effectively mitigate climate effects while preserving operational manageability.

Acknowledgment

This research was carried out as a part of the EU-Project RefMAP. RefMAP has received funding from the Horizon Europe program 2023 under grant agreement No 101096698. Tingting Ni was supported by the Swiss National Science Foundation.

References

[1] D. S. Lee, D. W. Fahey, A. Skowron, M. R. Allen, U. Burkhardt, Q. Chen, S. J. Doherty, S. Freeman, P. M. Forster, J. Fuglestvedt *et al.*, "The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018," *Atmospheric environment*, vol. 244, p. 117834, 2021.

[2] D. S. Lee, G. Pitari, V. Grewe, K. Gierens, J. E. Penner, A. Petzold, M. Prather, U. Schumann, A. Bais, T. Berntsen *et al.*, "Transport impacts on atmosphere and climate: Aviation," *Atmospheric environment*, vol. 44, no. 37, pp. 4678–4734, 2010.

[3] S. Dietmüller, S. Matthes, K. Dahlmann, H. Yamashita, A. Simorgh, M. Soler, F. Linke, B. Lührs, M. M. Meuser, C. Weder *et al.*, "A python library for computing individual and merged non-CO2 algorithmic climate change functions: CLIMaCCF V1. 0," *Geoscientific Model Development*, vol. 16, no. 15, pp. 4405–4425, 2023.

[4] A. Simorgh, M. Soler, D. González-Arribas, F. Linke, B. Lührs, M. M. Meuser, S. Dietmüller, S. Matthes, H. Yamashita, F. Yin *et al.*, "Robust 4D climate-optimal flight planning in structured airspace using parallelized simulation on GPUs: ROOST V1. 0," *Geoscientific model development*, vol. 16, no. 13, pp. 3723–3748, 2023.

[5] A. Simorgh, M. Soler, S. Dietmüller, S. Matthes, H. Yamashita, F. Castino, and F. Yin, "Robust 4D climate-optimal aircraft trajectory planning under weather-induced uncertainties: Free-routing airspace," *Transportation Research Part D: Transport and Environment*, vol. 131, p. 104196, 2024.

[6] A. Simorgh, M. Soler, D. González-Arribas, S. Matthes, V. Grewe, S. Dietmüller, S. Baumann, H. Yamashita, F. Yin, F. Castino *et al.*, "A comprehensive survey on climate optimal aircraft trajectory planning," *Aerospace*, vol. 9, no. 3, p. 146, 2022.

[7] F. Baneshi, M. Cerezo-Magaña, and M. Soler, "Integrating Non-CO2 climate impact considerations in air traffic management: Opportunities and challenges," *Transport Policy*, 2024.

[8] F. Baneshi, M. Soler, and A. Simorgh, "Conflict assessment and resolution of climate-optimal aircraft trajectories at network scale," *Transportation Research Part D: Transport and Environment*, vol. 115, p. 103592, 2023.

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[10] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3826–3839, 2020.

[11] A. Oroojlooy and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Applied Intelligence*, vol. 53, no. 11, pp. 13 677–13 722, 2023.

[12] S. Gu, J. Grudzien Kuba, Y. Chen, Y. Du, L. Yang, A. Knoll, and Y. Yang, "Safe multi-agent reinforcement learning for multi-robot control," *Artificial Intelligence*, vol. 319, p. 103905, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0004370223000516

[13] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 611–24 624, 2022.

[14] H. Kang, X. Chang, J. Mišić, V. B. Mišić, J. Fan, and Y. Liu, "Cooperative uav resource allocation and task offloading in hierarchical aerial computing systems: A mappo-based approach," *IEEE Internet of Things Journal*, vol. 10, no. 12, pp. 10 497–10 509, 2023.

[15] M. W. Brittain and P. Wei, "One to any: Distributed conflict resolution with deep multi-agent reinforcement learning and long short-term memory," in *AIAA Scitech 2021 Forum*, 2021, p. 1952.

[16] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[17] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of reinforcement learning and control*, pp. 321–384, 2021.

[18] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson, "Learning to communicate with deep multi-agent reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[19] F. Christianos, G. Papoudakis, M. A. Rahman, and S. V. Albrecht, "Scaling multi-agent reinforcement learning with selective parameter sharing," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1989–1998.

[20] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*. Springer, 2017, pp. 66–83.

[21] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International conference on machine learning*. PMLR, 2015, pp. 1889–1897.

[22] A. Ray, J. Achiam, and D. Amodei, "Benchmarking safe exploration in deep reinforcement learning," *arXiv preprint arXiv:1910.01708*, vol. 7, no. 1, p. 2, 2019.

[23] E. A. Group *et al.*, "Complexity metrics for ANSP benchmarking analysis," *EUROCONTROL, April*, 2006.