



# Scalable Multi-Agent Computational Guidance with Separation Assurance for Autonomous Urban Air Mobility

Xuxi Yang\* and Peng Wei\*,†  
Iowa State University, Ames, Iowa 50011

<https://doi.org/10.2514/1.G005000>

Electric vertical takeoff and landing vehicles are becoming promising for on-demand air transportation in urban air mobility (UAM). However, successfully bringing such vehicles and airspace operations to fruition will require introducing orders of magnitude more aircraft to a given airspace volume. Although there are existing solutions for communication technology, onboard computing capability, and sensor technology, the computation guidance algorithm to enable safe, efficient, and scalable flight operations for dense self-organizing air traffic still remains an open question. In this paper, a message-based decentralized computational guidance algorithm is proposed and analyzed for multiple cooperative aircraft by formulating this problem using multi-agent Markov decision process and solving it by Monte Carlo tree search algorithm. **A novel coordination strategy is introduced by using the logit level- $k$  model in behavioral game theory.** To achieve higher scalability, the airspace sector concept is introduced into the UAM environment by dividing the airspace into sectors, so that each aircraft only needs to coordinate with aircraft in the same sector. At each decision step, all of the aircraft will run the proposed computational guidance algorithm onboard, which can guide all the aircraft to their respective destinations while avoiding potential conflicts among them. For validation and demonstration, a free-flight airspace simulator that incorporates environment uncertainty is built in an OpenAI Gym environment. Numerical experiment results over several case studies, including the roundabout test problem, show that the proposed computational guidance algorithm has promising performance even with the high-density air traffic case.

## I. Introduction

### A. Motivation

THE increase in road traffic congestion in urban areas is a major concern for commuters and a burden on both the environment and the economy. This leads to the interest in urban air mobility (UAM), in which the electric vertical takeoff and landing (eVTOL) aircraft is able to alleviate transportation congestion by using three-dimensional airspace efficiently for passenger transport in personal commute or on-demand air taxi. A growing community of interest is forming for the concept of UAM, including NASA, Uber, Airbus, Honeywell, and many other entities around the globe [1–4]. Over a dozen companies, including Airbus, Bell, Embraer, Volocopter, and Aurora Flight Sciences, are building and testing their eVTOL aircraft to make it a reality.

However, successfully bringing the scalable UAM operations to fruition will require introducing orders of magnitude more aircraft to a given airspace volume, and it is estimated that there will be 23,000 aircraft flying major routes within the UAM network by 2035 [5]. Thus, the technical challenge is to provide concepts, technologies, and procedures that enable safe and efficient flight operations for the large number of eVTOL aircraft in the given airspace [6]. The Federal Aviation Administration's (FAA) Next Generation Air Transportation System program aims to modernize air traffic control (ATC) and aircraft systems to increase the capacity of the airspace [7]. However, the projected capacity increases over the 20 year period are expected to be no more than 50% [8,9], which is sufficient for the increased demand of commercial aircraft, but far below the requirement for the UAM air transportation system. In unmanned aircraft systems (UAS) traffic management (UTM), research efforts to increase the airspace capacity typically rely on aircraft being sufficiently equipped and

automated that they can operate relatively independently from the existing ATC system and are therefore not subject to its capacity limits [10,11].

In UAM, researchers have proposed structured airspace, where the eVTOL aircraft will fly according to fixed routes [12]. **In this paper, we consider a free-flight airspace framework** [13] because it was shown in previous work [14–17] that free flight with airborne separation is able to handle a higher air traffic density even in the presence of various uncertainties and delays. In addition, free flight can also increase fuel and time efficiency [18]. **In a free-flight framework, it is implied that aircraft will be responsible for self-separation assurance and conflict resolution** [19–21]. Removing the airway structure may make the process of detecting and resolving conflicts between aircraft more complex. However, a previous study [22] shows that free flight is potentially feasible because of enabling technologies, such as **Global Positioning Systems, data link communications (such as Automatic Dependent Surveillance-Broadcast [23]), the Next-Generation Airborne Collision Avoidance System (ACAS) [24], and powerful onboard computation.**

### B. Related Work

Decades of research have explored a variety of approaches for designing collision avoidance systems for both manned and unmanned aircraft, large commercial aircraft, and small unmanned aircraft. For conflict detection and resolution of commercial aircraft, there exist several surveys: Kuchar and Yang [25] give a comprehensive survey of air traffic conflict detection and resolution systems. Netjasov and Janic [26] provide a high-level outline in safety risk analysis, and a recent survey [27] presents a unified mathematical framework for air traffic conflict and collision definitions and methods to estimate the probability of conflict and collision.

Currently, **the Traffic Alert and Collision Avoidance System (TCAS)** is the only widely deployed aircraft collision avoidance system, which is required on all large transport aircraft in the world. If the system predicts that the intruder will penetrate a predefined safety buffer, the system will issue a resolution advisory to the pilot to adjust the vertical speed of the aircraft. Recent work on formulating the problem of collision avoidance as a partially observable Markov decision process (POMDP) has led to the development of the ACAS X family of collision avoidance systems [24,28]. The version for manned aircraft, ACAS Xa, is expected to become the next international standard for large commercial transport and cargo aircraft. Both TCAS and ACAS are designed to resolve

Received 5 December 2019; revision received 1 March 2020; accepted for publication 8 March 2020; published online 8 May 2020. Copyright © 2020 by Xuxi Yang and Peng Wei. Published by the American Institute of Aeronautics and Astronautics, Inc., with permission. All requests for copying and permission to reprint should be submitted to CCC at [www.copyright.com](http://www.copyright.com); employ the eISSN 1533-3884 to initiate your request. See also AIAA Rights and Permissions [www.aiaa.org/randp](http://www.aiaa.org/randp).

\*Graduate Research Assistant, Department of Aerospace Engineering; [xuxiyang@iastate.edu](mailto:xuxiyang@iastate.edu). Student Member AIAA.

†Assistant Professor, Department of Aerospace Engineering; [pwei@iastate.edu](mailto:pwei@iastate.edu). Senior Member AIAA.

one-on-one conflicts between aircraft with vertical maneuvers. The difference between them is TCAS uses fixed rules to resolve the conflicts, whereas ACAS uses a probabilistic model [solving a Markov decision process (MDP) problem with discrete state space], which leads to a better performance than TCAS. For multiple aircraft case, the Autonomous Operations Planner developed by NASA [29] is a flexible and powerful prototype of a flight-deck automation system to support self-separation of aircraft while en route. It incorporates a variety of algorithms that provide flight crew support for strategic and tactical conflict resolutions and conflict-free trajectory planning while meeting route constraints and avoiding airspace hazards. In this paper, we investigate how to resolve conflicts for multiple aircraft and guide the aircraft to their destinations through a series of actions by formulating this problem as an MDP and solving it using the Monte Carlo tree search (MCTS) algorithm.

For UTM, NASA's UTM project aims to enable the increasing number of low-altitude small UAS (sUAS) operations in uncontrolled airspace [30], specifically to enable safe and efficient en route UAS operations for civilian and public applications. The Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems software architecture (ICAROUS), being developed as part of the UTM project, will provide highly assured core software modules for building safety-centric autonomous unmanned aircraft applications [31]. A recent implementation known as Detect and Avoid Alerting Logic for Unmanned Systems (DAIDALUS) [32] is the basis for the ICAROUS software architecture. The core logic of DAIDALUS consists of 1) definition of self-separation threshold and well-clear violation volume, 2) algorithms for determining if there exists a potential conflict between aircraft pairs within a given look-ahead time, and 3) a determine-processing functionality that provides maneuver guidance and alerting logic.

Although research on a UTM system [30] for sUAS operating at low altitudes is relevant for UAM, it provides services appropriate for sUAS that do not always easily extend to the UAM environment [6]. For example, the risk of human injuries in the collision of two sUAS is very low [33], whereas larger vehicles with humans onboard will present significantly higher risks and the safety standards will have to be significantly enhanced for eVTOL aircraft in UAM over sUAS in UTM. Also, sUAS have the freedom to take off and land nearly anywhere, whereas eVTOL aircraft in UAM will be restricted to a network of vertiports, and therefore require scheduling and spacing services in vertiport terminal airspace.

In this paper, we mainly focus on guidance and conflict resolution systems for these new entrants (such as eVTOL UAM aircraft) in the metropolitan airspace. In the applications for these new entrants, the existing work can be categorized based on the following criteria. We will discuss the related work based on the following categories:

1) Centralized/decentralized [34]: centralized algorithms solve the problem by a central supervising controller, whereas decentralized algorithms solve the problem by each aircraft individually.

2) Planning/reacting [35]: the planning approach generates feasible or even optimal paths ahead of execution, whereas the reacting approach typically uses an onboard collision avoidance system to respond to obstacles and other vehicles.

3) Cooperative/noncooperative: cooperative algorithms allow communication among aircraft, whereas the aircraft cannot communicate with each other in noncooperative algorithms.

In centralized methods, the conflicts between aircraft are resolved by a central supervising controller, where most of them require planning ahead of time. Under such scenarios, the state of each aircraft, the obstacle information, and the trajectory constraint (such as required times of arrival and restricted airspace area) are known to the central controller, thus centralized methods are always cooperative. The central controller in return designs the whole trajectory for each aircraft predeparture or en route, typically by formulating it as an optimal control problem. These methods can be based on semidefinite programming [36], nonlinear programming [37,38], mixed-integer linear programming [39–42], mixed-integer quadratic programming [43], sequential convex programming [44,45], evolutionary techniques [46,47], and particle swarm optimization [48]. Besides formulating this problem using optimal control framework, computational geometry methods, such as visibility graph [49] and Voronoi diagrams [50],

can also handle the path planning problem for aircraft without modeling detailed vehicle dynamics. However, calculating the exact solution for such computational-geometry-based robot motion planning will become intractable [51] when the state space becomes large or high dimensional. To address this issue, sample-based planning algorithms are proposed, such as probabilistic roadmaps [52], RRT [53], and RRT\* [54,55]. These centralized methods often generate the whole trajectories for agents. However, as the number of aircraft grows (in multi-agent case), the computation time of these methods typically scales exponentially. Moreover, these centralized planning approaches typically need to be rerun, as new information in the environment is updated (e.g., a new aircraft enters the airspace, or one aircraft failed to execute its planned trajectory).

On the other hand, decentralized methods scale better with respect to the number of agents and are more robust because they are not vulnerable to a single point of failure. However, because the agents act only on local information, global optimality of a decentralized control policy is often hard to achieve [56]. In decentralized methods, all the conflicts are resolved by each aircraft individually. Decentralized methods can be cooperative and noncooperative. Researchers have proposed several algorithms under the case where the communication between aircraft can be successfully established (cooperative with communication) [57]. Algorithms in [58,59] are based on message-passing schemes, which resolve local (e.g., pairwise) conflicts without needing to form a joint optimization problem between all members of the team. In [34], every agent is allotted a time slot to compute a dynamically feasible and collision-free path using mixed-integer linear programming. In [60], the authors recast the global optimization problem as several local problems, which are then iteratively solved by the agents in a decentralized way. In the decentralized model predictive control approach [61], the aircraft solve their own subproblem one by one and send the action to other subsystems through communication.

There are also scenarios where communication cannot be reliably established (noncooperative) and the aircraft will take action at each time step based on the sensor information. Many works fall in this category: model predictive control [62,63] can be used to solve the collision avoidance problem, but the computation load is relatively high. Potential field method [64] is computationally fast. However, a navigation function [65] is required to deal with the local minima problem and make it a complete path planner [66,67], which involves discretizing the state space. With the help of machine learning and reinforcement learning [68–72], collision avoidance algorithm (without trajectory replanning to the destination) can have a promising performance, but the data collection and model training part are expensive. Using the MCTS algorithm to solve this problem [73] does not need model training and the algorithm can finish in any predefined computation time, but the aircraft can only adopt several discretized actions at each time step. In [74], the authors proposed a sample-based POMDP approximation algorithm that can run onboard for continuous state and observation spaces, which can find the optimal action using the branch-and-bound method. However, computation time was the most limiting factor in this work. Geometry-based algorithms [75–78] can be also applied for collision avoidance problem and the computation time only grows linearly with the increasing number of aircraft. The drawback of these geometric approaches is that they cannot look ahead for more than one step since they only pay attention to the current action and do not take account of the effect of subsequent actions, thus the outcome can be local optimal in the view of the global trajectory.

In this paper, under the free-flight framework, we propose a computational guidance algorithm with a separation assurance capability, which is a message-passing-based decentralized, reacting, and cooperative algorithm. We formulate this computational guidance problem as a multi-agent MDP (MMDP) and solve the formulated MMDP using the MCTS algorithm that can run onboard the aircraft. In fact, the proposed algorithm in this paper can be either centralized (where a centralized controller is responsible for gathering aircraft state information and issuing action advisories to all of the aircraft) or decentralized (where the algorithm runs onboard the aircraft and aircraft can coordinate with each other through wireless

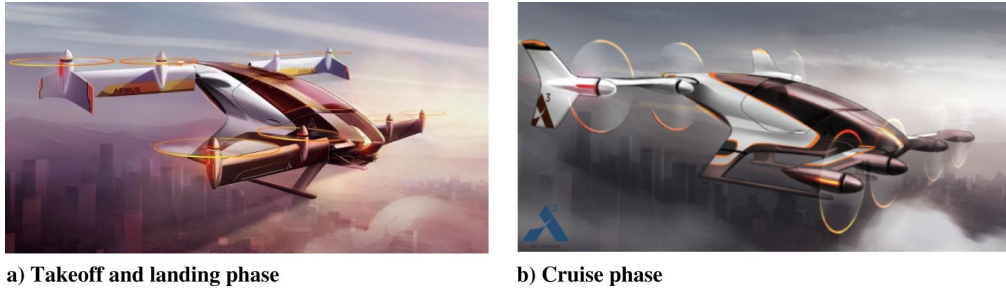


Fig. 1 Airbus Vahana aircraft during the takeoff phase and cruise phase [87].

communication). We described both centralized and distributed cases in detail in our recent paper [79]. In this paper, we focus on the distributed case. There are similar works using MDP formulation, which solve this problem offline in the predeparture phase [24,69,80] or online during the en route phase [81]. Offline solvers require large computation time up front to compute the optimal policy for the full state space and discrete MDP formulations. Offline methods are typically not adaptive to changes in the environment because the policy is determined ahead of time. Also, the state space of many problems is too large to adequately represent as a finite set of enumerable states. Comparing with offline methods, by using longer onboard computation time, onboard methods address the shortcomings of offline methods by planning only for the current state and a small number of possible plans. Because onboard algorithms only need to plan for the current state that can take any continuous value, state discretization of MDP formulation is not required. Onboard algorithms are also able to account for changes in the environment because they are executed once at each decision point, allowing for updates between these points. There is also POMDP formulation for this problem, which aims to consider the state uncertainty due to the sensor noise [74], in which the authors use a search algorithm to solve this POMDP. The major difference between this paper and our work is that they use depth-first search and **do not consider the state transition uncertainty**, whereas we use the robust MCTS algorithm to handle the dynamic model uncertainty through simulations.

### C. Airspace Sectorization

Airspace sectorization is a concept widely used in commercial aviation. According to the FAA [82], the airspace sector is defined to be an airspace area with predefined horizontal and vertical dimensions for which a controller or a group of controllers has ATC responsibility, normally within an air route traffic control center or an approach control facility. Sectors are established based on predominant traffic flows, altitude strata, and controller workload.

**Airspace sectors can be created to deal with the high demand for aircraft traffic** [83]. For example, in times when there are high levels of air traffic, more sectors may be opened with more controllers allocated to manage the aircraft within an area of airspace. This is done to maintain safety as a controller can only manage a certain number of aircraft at one time. In this paper, we use airspace sectors to **reduce the computation time** for the proposed computational guidance algorithm.

Although the airspace sectorization can help reduce the computation time by distributing the workload to several sectors, it also introduces complexity by requiring additional coordination between aircraft and sector controllers, especially when an aircraft flies across more than one airspace sector.

**Note that the airspace sectorization also introduces additional hand-off conflicts between aircraft near the boundary between (at least) two sectors** [84], because the adjacent sectors might have different plans to resolve the conflict and not able to see the traffic situation in the neighboring sector. This type of loss of separation deserves special attention because of the hand-off between two sectors. Several typical scenarios that could cause hand-off conflicts include poor or missing coordination/communication between sectors, aircraft flying along the sector boundary, and different minimum separation standards used in adjacent sectors. Also, adverse weather

avoidance, communication equipment failure, high controller workload, transfer of control too early/too late, and sector skipping can contribute to conflicts between aircraft as well. For more details, readers can refer to [84].

In this paper, we will design a novel mechanism for airspace sectorization to resolve the hand-off conflict mentioned previously.

### D. Contributions

In this paper, we propose a message-passing-based decentralized computational guidance algorithm with a separation assurance capability that can scale to multiple cooperative aircraft, where we formulate this problem as an MMDP and solve this formulated MMDP using MCTS algorithm. **We use a logit level- $k$  model for the multi-aircraft coordination based on the message-passing mechanism through wireless communication. To achieve higher scalability, we introduce the airspace sector concept into the UAM environment by dividing the airspace into sectors, so that each aircraft only needs to coordinate with aircraft in the same sector.** A high-density free-flight airspace simulator in the OpenAI Gym environment is built to validate and demonstrate the performance of the proposed algorithm. Through numerical experiments over several case studies in the free-flight simulator with environment uncertainty, the proposed algorithm shows promising performance. Additionally, we also performed a stress test on the roundabout test problem, which consists of making a certain number of aircraft fly to the diametrically opposed point at a common speed on an annulus.

In the free-flight airspace simulator, the aircraft dynamics are modeled based on the tandem tilt-wing eVTOL (Airbus Vahana) from Airbus A<sup>3</sup> [85], which has flown over 80 full-scale test flights [86]. Figure 1 shows the takeoff and landing phase and cruise phase of Vahana aircraft [87]. **In this paper, we restrict our scope to the cruise phase of this aircraft in en route airspace.** For the scheduling and spacing services in the vertiport terminal airspace (arrival and departure management), readers can refer to [88–90].

Overall, this research proposes a potential solution to integrate the power of onboard aircraft autonomy and the free-flight concept for airspace operations to enable safe and efficient flight operations in high-density urban air traffic.

The structure of the paper is as follows: in Sec. II, the background of MDP, MCTS, and MMDP will be introduced. In Sec. III, the description of the problem and its mathematical formulation of MMDP are presented. Section IV presents the designed MCTS algorithm to solve this multi-agent problem. The numerical experiment and results are shown in Sec. V. Section VI is the conclusion.

## II. Background

In this section, we briefly review the background of the MDP and MCTS, as well as the MMDP.

### A. Markov Decision Process

Since the 1950s, MDPs [91] have been well studied and applied to a wide area of disciplines [92–94], including robotics [95,96], automatic control [97], economics, and manufacturing. In an MDP, the agent may choose any action  $a$  that is available based on current state  $s$  at each time step. The process responds at the next time step by



moving into a new state  $s'$  with certain transition probability, and gives the agent a corresponding reward  $r$ .

More precisely, the MDP includes the following components:

- 1) The state space  $\mathcal{S}$  consists of all the possible states.
- 2) The action space  $\mathcal{A}$  consists of all the actions that the agent can take.
- 3) Transition function  $\mathcal{T}(s_{t+1}|s_t, a_t)$  describes the probability of arriving at state  $s_{t+1}$ , given the current state  $s_t$  and action  $a_t$ .
- 4) The reward function  $\mathcal{R}(s_t, a_t, s_{t+1})$  decides the immediate reward (or expected immediate reward) received after transitioning from state  $s$  to state  $s'$ , due to action  $a$ . In general, the reward will depend on the current state, current action, and the next state. However, the reward function may only depend on the current state  $s_t$ , which will be the case in this paper.
- 5) A discount factor  $\gamma \in [0, 1]$  decides the preference on immediate reward vs future rewards. Setting the discount factor less than 1 is also beneficial for the convergence of cumulative reward.

In an MDP problem, a policy  $\pi$  is a mapping from the state to one specific action (known as deterministic policy):

$$\pi: \mathcal{S} \rightarrow \mathcal{A} \quad (1)$$

The goal of MDP is to find an optimal policy  $\pi^*$  that, if followed from any initial state, maximizes the expected cumulative rewards over all the future steps:

$$\pi^* = \arg \max_{\pi} \mathbb{E} \left[ \sum_{t=0}^{T-1} R(s_t, a_t, s_{t+1}) | \pi \right] \quad (2)$$

$Q$  function and value function are two important concepts in MDP. The optimal  $Q$  function  $Q^*(s, a)$  means the expected cumulative reward received by an agent starting in state  $s$  and picks action  $a$ , and then will behave optimally afterward. Therefore,  $Q^*(s, a)$  is an indication of how good it is for an agent to pick action  $a$  while being in state  $s$ . The optimal value function  $V^*(s)$  denotes the maximum expected total reward when starting from state  $s$ , which can be expressed as the maximum of  $Q^*(s, a)$  over all possible actions:

$$V^*(s) = \max_a Q^*(s, a) \quad \forall s \in \mathcal{S} \quad (3)$$

## B. Monte Carlo Tree Search

Monte Carlo tree search is a method for finding optimal decisions in a given domain by taking random samples in the decision space and building a search tree according to the results [98,99]. It has already had a profound impact on artificial intelligence approaches for domains that can be represented as trees of sequential decisions, particularly games and planning problems [100,101], including the current state-of-the-art computer program AlphaZero in the game of Go [102].

The basic MCTS process is conceptually easy to understand, in which a tree is built in an incremental and asymmetric manner, as shown in Fig. 2 (from [99]). For each iteration of the algorithm, a tree policy is used to find the most urgent node of the current tree. The tree policy attempts to balance considerations of exploration and exploitation. Exploration means to look in areas that have not been well

sampled yet and exploitation means to look in areas that appear to be promising. A simulation is then rolled out from the selected node and the search tree updated according to the result. This involves the addition of a child node corresponding to the action taken from the selected node and an update of the statistics of its ancestors. Moves are made during this simulation according to some default policy, which, in the simplest case, is to make uniformly random moves. A great benefit of MCTS is that the values of intermediate states do not have to be evaluated, as for depth-limited minimax search, which greatly reduces the amount of domain knowledge required. Only the value of the terminal state at the end of each simulation is required.

## C. Multi-Agent Markov Decision Process

Comparing with previous work [73], in which the algorithm can only control one single aircraft, in this paper, **we focus on how the MCTS algorithm can scale to multiple cooperative agents**.

To extend MDPs to multi-agent settings, researchers [103–105] have introduced MMDPs, which allow for representing sequential decision-making problems in cooperative multi-agent settings. Similar to MDP, MMDP is defined as a tuple  $\langle n, \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$  [106], in which

- 1) The scalar  $n$  is the number of agents in the whole system.
- 2)  $\mathcal{S}$  is the set of states  $s$ .
- 3)  $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_n$  denotes the set of joint actions, where  $\mathcal{A}_i$  is the set of individual actions of agent  $i$ .
- 4)  $\mathcal{T}$  is the transition function, which gives the probability  $\mathcal{T}(s_{t+1}|s_t, a_t)$  that the process moves to state  $s_{t+1}$  when the agents execute the joint action  $a \in \mathcal{A}$  from state  $s_t$ .
- 5)  $\mathcal{R}(s_t, a_t, s_{t+1})$  is the reward obtained when the process state changes from  $s_t$  to  $s_{t+1}$  under the influence of joint action  $a_t$ .

Although MMDP can model the interactions among multiple agents, there are also many challenges in MMDP [107]:

- 1) The curse of dimensionality will cause exponential growth of the discrete state–action space in the number of state and action variables. For example, assume we have 10 agents, and each agent has three actions at each time step, then there will be  $3^{10}$  different action combinations to consider at each time step, and each action could be the optimal one.

- 2) **Nonstationarity arises in multi-agent systems because each agent is facing a moving-target learning problem: the best policy changes as the other agents' policies change.** For example, as shown in Fig. 3, without knowing the action of the other aircraft, we cannot tell which action is optimal because each action could lead to a Loss of Separation (LOS) event between them. **Also the optimal action of an aircraft will be affected if other aircraft changed their actions.**

Most of the literature studied multi-agent systems in stochastic environments with a focus on Nash equilibrium [108] or long-term stable behaviors. But, when we are dealing with real-time decision-making systems, information about Nash equilibrium is not always helpful [109]. First of all, it may be unclear which equilibrium to adopt if there are many different equilibria in the system. For systems with only one equilibrium, it may be difficult to compute the Nash equilibrium when the computation time is limited [110]. An area known as behavioral game theory [111] aims to model agents that are limited in the number of steps of strategic look ahead when the decision time is limited. Many different behavioral models exist, but the logit level- $k$  model [112,113] has become popular recently

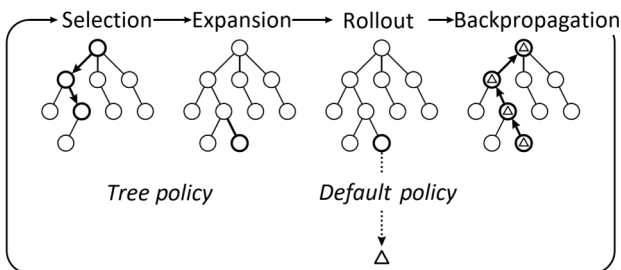


Fig. 2 One iteration of general MCTS approach [99].

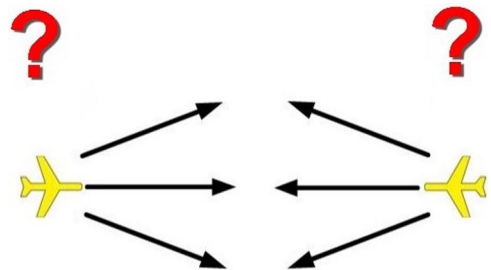


Fig. 3 Aircraft coordination is necessary for action selection.

and tends to work well in practice. In the logit level- $k$  model, an agent at logit level- $k$  assumes all of the other agents follow logit level- $(k - 1)$  strategy. In this way, the actions of other agents become fixed; thus, the tree search process can avoid the action explosion issue in the multi-agent case.

In this paper, we use a variant of the logit level- $k$  model to make decisions for all of the agents in a fast, dynamic, and uncertain environment, where real-time onboard decisions are needed.

### III. Problem Formulation

#### A. Problem Statement

The goal of this research was to develop a distributed algorithm that runs on each aircraft to provide tactical guidance commands. These commands will help each aircraft arrive at their respective destinations while avoiding potential LOS events between them during the flight. In this paper, we assume a free-flight concept of operations (ConOps), where each flight's trajectory is free of airspace constraint. Here, we investigate the feasibility of the free-flight ConOps and push its safety limits with new tools from MMDP and MCTS. However, we would like to identify other approaches and ConOps, such as predeparture flight planning for strategic deconfliction, and/or implementing structured airspace to ensure higher level of safety. In the MMDP formulation, at each decision point, the vehicle's action is decided directly from the state, which incorporates all the information (the position and velocity of all the intruder aircraft and the destination/goal of ownship) to decide which action is optimal for the corresponding state.

When controlling the aircraft, only horizontal actions are considered in this paper, which means all the aircraft will be flying at the same altitude, and this problem can be solved in two dimensions. This assumption makes it possible to incorporate multiple flight levels to deal with the high-density air traffic in UAM.

The objectives for this specific MMDP problem are twofold: the first is to avoid potential LOS events among all the moving aircraft, and the second is to guide all the aircraft to their destinations as quickly as possible during the flight. Therefore, the reward function needs to capture both two objectives.

Based on the aforementioned description, this problem will be mathematically formulated as an MMDP problem in the next subsection.

#### B. MMDP Formulation

The MMDP formulation in this paper is similar to our previous work [73], and here we will briefly introduce it.

##### 1. State Space

The state includes the necessary information for the algorithm to issue actions to all the aircraft: the position, speed, heading angle, and goal position for all of the aircraft. More specifically, the state for one aircraft is  $(x, y, v, \psi, g_x, g_y)$ , where  $(x, y)$ ,  $v$ ,  $\psi$  are the position, speed, and heading angle for the aircraft, and  $(g_x, g_y)$  is the goal position for this aircraft. Stacking the information for all of the controlled aircraft, the state space for the MMDP becomes an  $n \times 6$  matrix, where  $n$  is the number of aircraft and each row of the matrix represents the information for one aircraft.

Note here that the state space is continuous; for example, all the variables of a state can take continuous values. In general, for an MDP with continuous state variables, it is not clear how to best represent the policy, because it is impossible to enumerate all possible state-action mappings. For previous MDP-based algorithms to solve conflict avoidance problems, some possible approaches to represent the policy include using a grid-based discretization of the state space  $\mathcal{S}$  and the action space  $\mathcal{A}$  [28,69], or using policy compression techniques [114,115]. The advantage of the MCTS algorithm is that it does not need to discretize the state space. For each state, the MCTS algorithm will generate action onboard for the aircraft to follow in real time.

##### 2. Action Space

At each time step (5 s), the aircraft can choose to turn its heading at a certain rate. More precisely, the advisory of heading angle for each aircraft constitutes the action set  $\mathcal{A} = \{-5, 0, +5 \text{ deg/s}\}$ , where positive corresponds to right turn and negative to left turn. The changing rate of the heading angle is determined assuming the aircraft is flying with cruise speed at 190 km/h [86] and banking angle at 25 deg. (The banking angle limit is chosen to be 25 deg for the passenger comfort consideration.) At each time step, the proposed algorithm will run onboard to issue one action from the action set for the aircraft based on the current state. After the algorithm running, the aircraft will maintain the issued action during this time step.

It is natural to consider extending the set of actions (conflict resolution advisories) to include more options, such as vertical resolution and speed resolution. However, using the MCTS algorithm to calculate the optimal action will be more time consuming with the extended action space because the tree size will grow exponentially with the number of actions. Because computation time is an important factor for the real-time onboard algorithm, some techniques are necessary for extending action space in future steps, such as truncated Monte Carlo search algorithms [116], progressive strategies for MCTS [117–120], or using a policy network to narrow down the search to high-value actions [100]. In this paper, we use this defined action space to keep our scope more focused.

##### 3. Dynamic Model

Based on the current state and current action, Dubins kinematic model will be used to compute state transition for each aircraft:

$$\dot{x} = v \cos \psi \quad (4)$$

$$\dot{y} = v \sin \psi \quad (5)$$

$$\dot{\psi} = a_{\psi} \quad (6)$$

where  $v$  is the cruise speed,  $\psi$  is the heading angle, and  $a_{\psi}$  is the selected action describing the changing rate of heading angle for one aircraft. Following the aircraft performance data from Airbus Vahana [86], the cruise speed of the aircraft is set to 190 km/h, and we restrict the speed to be in between  $v_{\min} = 165 \text{ km/h}$  and  $v_{\max} = 220 \text{ km/h}$  [85,86] because there is uncertainty in the speed.

After an aircraft executes an advisory, the aircraft speed is held constant between decision stages with uncertainty, which is modeled as a Gaussian distribution centered on the aircraft's cruise speed with a standard deviation of 5 m/s. The changing rate of the heading angle distribution is centered on the resolution advisory with a standard deviation of 2 deg/s. The noises here aim to account for the uncertainties in the environment and aircraft dynamics. In previous works [69,121], the uncertainties are modeled as Gaussian distributions with a standard deviation of 2 m/s for speed and 3 deg for heading angle when the banking angle is less than 25 deg. In this paper, we increase the standard deviation for speed to 5 m/s and lower the standard deviation for change of heading angle to 2 deg/s to better model the uncertainties because Vahana aircraft is flying at a higher speed.

##### 4. Reward Function

The focus of our computational guidance system is to achieve the dual objectives of maintaining safety while guiding the aircraft to their destinations as quickly as possible. These objectives are captured in a reward function composed of a sum of the reward function for each individual aircraft.

For the consideration of safety, the LOS event is defined to be when the distance of two aircraft is less than a minimum separation distance  $r^{\min} = 0.5 \text{ n mile}$  [122]. This separation standard was chosen using the definition of well clear for UAS according to Cook and Brooks [123]. For large UAS in high-altitude airspace, the horizontal miss distance (HMD) is defined to be 0.66 n mile. For sUAS (55 lb vehicle or less) in low-altitude controlled airspace around airports, the horizontal separation is set to be an HMD of 0.36 n mile. Using those

values as reference, the nominal spatial separation standards picked for this UAM application are set to 0.5 n mile horizontally. This value is tighter than UAS standards because it is assumed that enhanced equipage capabilities will be installed on the eVTOL aircraft [122].

Based on the aforementioned separation requirements, we define the following two different types of states:

1) The distance between two aircraft is less than  $r^{\min}$  (referred to as an LOS state in the following).

2) The aircraft reaches the goal position (referred to as a goal state in the following).

With the two types of states defined previously, the reward function for one aircraft is defined as follows:

$$r(s) = \begin{cases} 1, & \text{if } s \text{ is goal state,} \\ 0, & \text{if } s \text{ is LOS state,} \\ 1 - \frac{d(o,g)}{\max d(o,g)}, & \text{otherwise} \end{cases} \quad (7)$$

where  $d(o, g)$  denotes the distance from the aircraft to its goal position;  $\max d(o, g)$  is the maximum distance from an aircraft to its goal position, which is the diagonal distance of the map. (If the map has an irregular convex shape, we can use the diameter of this convex shape.) In this way, if an aircraft is not at LOS state (which has reward 0), this aircraft will get a positive reward between 0 and 1, depending on how far this aircraft is from the goal state.

The reward function for one aircraft is normalized to range [0, 1] for the following reason. In MCTS, the algorithm will keep selecting the action with maximum value:

$$\bar{r}_j + 2C \sqrt{\frac{2 \ln n}{n_j}} \quad (8)$$

where  $\bar{r}_j$  is the mean reward value for action  $j$ ,  $n$  is the number of times the current state has been visited, and  $n_j$  is the number of times action  $j$  has been selected during the process for building the search tree. In Eq. (8), the first term describes the average value for an action from previous information, the second term measures the uncertainty of this action, and the coefficient  $C$  can balance these two terms. It should be noted that the value of  $C$  depends on the reward scale  $\bar{r}_j$ . The reward function is normalized to range [0, 1] because, for this reward range, the value of  $C = 1/\sqrt{2}$  was shown by Kocsis et al. to satisfy the Hoeffding's inequality [124]. With a reward range different from [0, 1], a different value of  $C$  will be needed.

After we have the reward function definition for one aircraft, the reward function for the MMDP is defined to be the sum of reward for each aircraft:

$$R(s) = \sum_i r_i(s) \quad (9)$$

where  $r_i(s)$  is the reward function for aircraft  $i$  defined in Eq. (7).

With this reward setting, when we solve the formulated MMDP by maximizing the reward, all of the aircraft will try to select action leading to a state that is closer to the goal position (which has positive reward) and avoid any LOS states (which has 0 reward). Because non-LOS states are always preferred than LOS states in the aforementioned reward setting, we do not need to introduce a penalty (such as a negative reward) for LOS states.

#### IV. Solution Method

In this paper, we will use the most popular algorithm in the MCTS family, the upper confidence bound for trees (UCT), to solve the MMDP problem formulated previously. The details of UCT algorithm implementation for the single aircraft case can be found in [73]. As formulated in Sec. III, computing the global optimal solution is impractical for more than a few aircraft because the state space will grow linearly and the action space will grow exponentially with the increasing number of aircraft. Therefore, we use a variant of logit level- $k$  model [112,113] to solve the aforementioned issue. In the logit level- $k$  model, an agent at logit level- $k$  assumes all of the other

agents follow logit level- $(k-1)$  strategy. In this way, the actions of other agents become fixed; thus, the tree search process can avoid the action explosion issue in the multi-agent case.

In the logit level- $k$  model, a level 0 agent selects actions randomly by following a uniform distribution. A level 1 agent assumes that all the other agents adopt level 0 strategies and selects actions according to the logit distribution:

$$P(a_i) \propto e^{Q_1^*(s, a_i)} \quad (10)$$

where  $Q_1^*(s, a_i)$  is the  $Q$  function for the state-action pair  $(s, a_i)$  of a level 1 agent assuming other agents follow level 0 strategies. A level- $k$  agent assumes that the other agents adopt level- $(k-1)$  strategies and select their own actions according to Eq. (10).

In this paper, we use a deterministic variant of the logit level- $k$  model. More specifically, a level 0 aircraft selects the action to fly straight, and a level 1 aircraft assumes other aircraft adopt the level 0 strategy and selects action  $a^*$  with maximum  $Q$  value:

$$a^* = \arg \max_{a_i} Q_1^*(s, a_i) \quad (11)$$

where the  $Q$  function  $Q_1^*(s, a_i)$  is approximated from the MCTS algorithm.

The deterministic logit level- $k$  model presented previously updates the action for the next level in a synchronous way; for example, the aircraft at the same level do not know the actions among each other. As shown in Fig. 3, knowing the action of other aircraft is helpful for the action selection; therefore, in this paper, we update the action for the next level in an asynchronous way.

More specifically, suppose currently we have  $n$  aircraft flying in the air. Then, at the beginning of the search algorithm when all the aircraft are at level 0, the joint actions  $\mathbf{a}$  for all aircraft are initialized to 0 at first:

$$\mathbf{a} = \{a_1, a_2, \dots, a_n\} \quad (12)$$

where  $a_i$  is the action for aircraft  $i$ , and  $a_1 = a_2 = \dots = a_n = 0$  deg/s.

Then, starting from the first aircraft, each aircraft will run the algorithm onboard by building a search tree, assuming all of the other aircraft will take action according to the joint action  $\mathbf{a}$  and follow the dynamic model described in Sec. III.B.3. The searching process is similar to the process described in [73], and the difference is all aircraft can turn according to the joint action.

Next, assume the tree search result for the first aircraft is  $a_1^*$ , then the first aircraft will send this action information  $a_1^*$  to all the remaining aircraft through wireless communication, and the joint action will be updated as follows:

$$\mathbf{a} = \{a_1^*, a_2, \dots, a_n\} \quad (13)$$

Note, here, only the first aircraft is at level 1 and all the other aircraft are still at level 0. Then, after receiving the action information from the first aircraft, the second aircraft begins running the MCTS algorithm onboard assuming all of the aircraft are following the most recent joint action. This process will iterate over all the aircraft until all the  $a_1, a_2, \dots, a_n$  are updated, when all the aircraft are at level 1. We can keep updating the actions for all of the aircraft to higher levels. Because numerical experiment results do not show much performance improvement for level 2 compared with level 1, for the consideration of computation time, in this paper, we stop this process after reaching level 1. After we have the level 1 joint action, all of the aircraft will execute this joint action for five time steps, after which all of the aircraft will run the algorithm again to generate new joint action.



## V. Numerical Experiments

### A. Simulator

To test the performance of the proposed algorithm, a simulator was built in Python, where multiple aircraft can fly freely in the two-dimensional en route airspace above New York City. We envision there will be multiple altitude levels, where the eVTOL aircraft are operated. In the scope of this paper, we only focus on one altitude level (a two-dimensional environment).

To validate the performance of this algorithm in real-world applications, we will simplify the UAM network by following the generic city model presented in [125,126]. In this generic city model, seven vertiports are distributed in a “six around one” hexagonal pattern. As shown in Fig. 4, vertiport 1 is in the center of the hexagon and located equidistant from the other six vertiports at a distance of 16 km, which will cover the main congestion area of New York City. Overlays of the vertiport network are shown on a Google map image of New York in Fig. 4, which shows typical New York traffic on a Friday at 1700 hrs [127].

In real-world scenarios, the arrival and departure routes of large, conventional aircraft in terminal airspace near airports [128], the presence of flight restrictions and restricted airspace [129], and the noise of the eVTOL vehicles [2] may all limit the UAM airspace shape and restrict the eVTOL operations between vertiports. The proposed algorithm in this paper does not work under such scenarios yet, especially when the restricted airspace is in an irregular shape. A direction of future work would be to adapt the proposed algorithm to incorporate various airspace restrictions to make it more practical.

### B. Airspace Sectorization

In the numerical experiment, the result shows that the computation time for the algorithm grows linearly with the increasing number of aircraft, which makes it intractable for real-time guidance when the number of aircraft is over 15. To mitigate this issue, in this paper, we introduce the concept of airspace sectorization to reduce computation time for the proposed algorithm by distributing the coordination workload to different sectors. As shown in Fig. 5, the airspace with

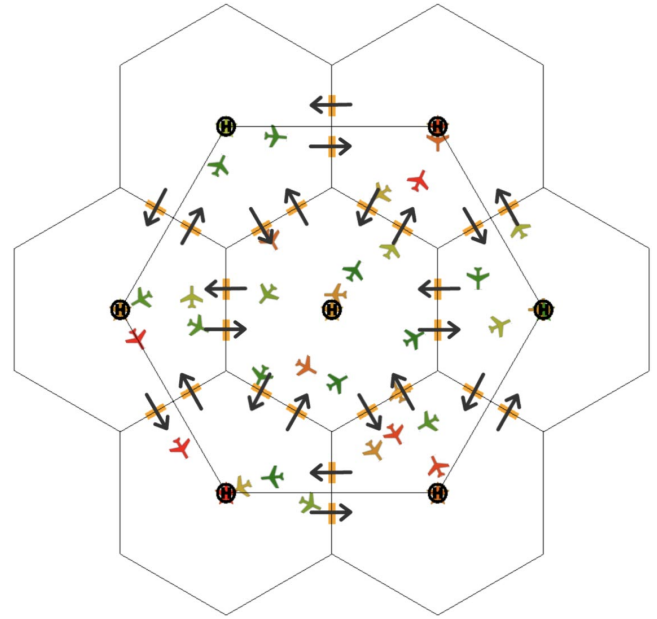


Fig. 5 Airspace sectorization can help reduce the computation time for the proposed algorithm.

seven vertiports is divided into seven hexagonal sectors and the center of each hexagonal sector is one vertiport. One advantage of this hexagonal sector configuration is that hexagons can form a tessellation of a two-dimensional airspace, which means this sector configuration can be easily extended to larger airspace.

In this sector setting, each aircraft only needs to coordinate with other aircraft in the same sector. More specifically, the aircraft in one sector only gather information of other aircraft in the same sector, and the decision-making process only depends on the gathered information. Every 5 s, aircraft in the same sector will simulate action and

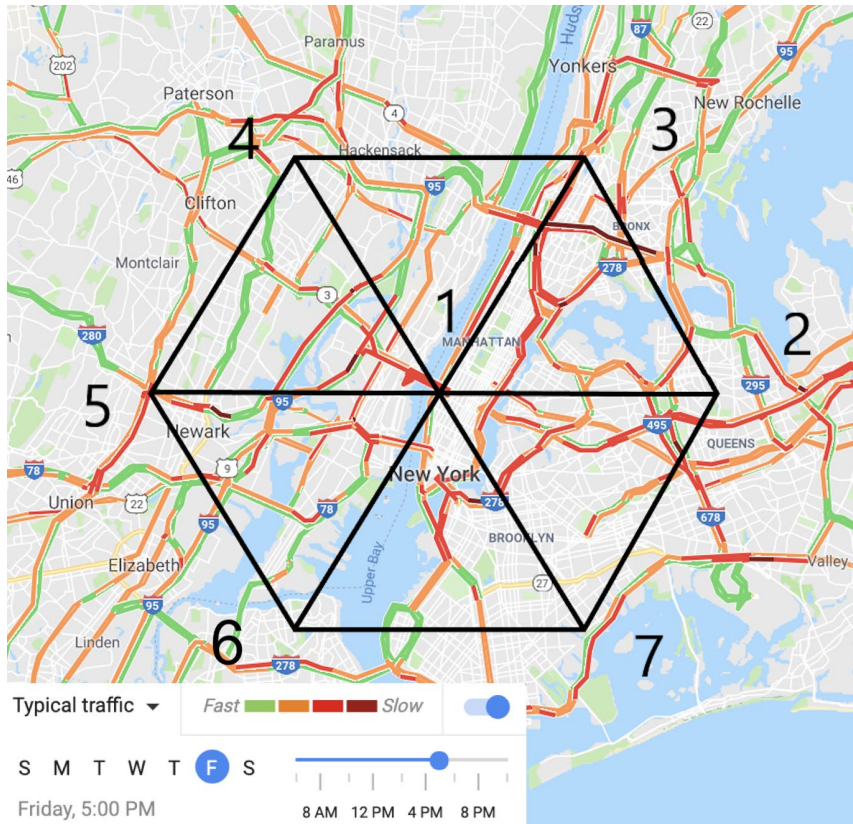


Fig. 4 Network of seven vertiports overlaid on New York City.

communicate the action information according to the default order (which can be based on the time when the aircraft enters this sector). With this sector setting, ideally, we can reduce the computation time by a factor of 7.

Although the sector configuration can help reduce the computation time for the proposed algorithm, it also introduces a new type of conflict called hand-off conflict: when the aircraft is crossing/flying near a boundary of the sector, there is higher chance for conflicts to happen because the aircraft does not know the aircraft information from other sectors. To resolve this problem, we introduce one-directional gates on the boundary of sectors, which is denoted as orange rectangles in Fig. 5 and the direction of the gate is denoted using arrows. The aircraft is only allowed to exit through the available gates, and exiting from other locations will be regarded as an LOS state with reward 0, which the algorithm will try to avoid. The width of the gates is designed to be 1800 m, so as to allow two aircraft passing the gates simultaneously. When an aircraft enters a new sector or takes off from a vertiport, it will be assigned an exit gate for this aircraft (if the goal vertiport of this aircraft is not in the current sector) by minimizing the total path:

$$\min \left\{ \sqrt{(p_x - e_x)^2 + (p_y - e_y)^2} + \sqrt{(e_x - g_x)^2 + (e_y - g_y)^2} \right\} \quad (14)$$

where  $(p_x, p_y)$ ,  $(e_x, e_y)$ ,  $(g_x, g_y)$  are the position for aircraft, exit gate, and the goal of the aircraft. After assigning the exit gate for this aircraft, the onboard algorithm will guide the aircraft toward its assigned exit gate.

To further reduce the risk of conflicts, the aircraft in one sector can also sense the aircraft information from other sectors that are close to this sector (if the distance between the aircraft and the sector is smaller than 1500 m), but the aircraft does not receive any action information (or flight intention) from them. **In summary, each aircraft will receive full information (state and action information) from other aircraft in the same sector, partial information (only state information) from other aircraft that are close to its own sector, and no information from all the remaining aircraft.**

### C. Parameter Settings

In the proposed algorithm, **there are two parameters that can impact the performance of the MCTS algorithm: the number of simulations and the search depth.** The number of simulations means the number of rollouts to simulate during the tree search process. Because there is uncertainty in the environment and aircraft dynamics, more rollouts will cover more cases and make the algorithm more robust. Search depth means how many steps to look ahead. A more detailed definition can be found in [73]. Typically, for the MCTS algorithm, a larger search tree (more simulations and deeper search depth) can lead to better performance of the algorithm, but needs longer computation time. **In a previous work [73], it is found that setting the number of simulations to 100 and search depth to 3 in this problem can have decent performance in terms of the number of LOS events/near midair collisions (NMACs), average en route flight time, and average onboard computation time.** Thus, in this paper, we adopt the same parameter setting. We also noticed that, when an aircraft is far from the other aircraft, a smaller tree is enough to find the good action. So to speed up the algorithm, when the distance of an aircraft to its closest aircraft is larger than 2 km, we set the number of simulations to 30 and search depth to 2.

### D. Case Studies and Results

Based on the aforementioned simulator setting, we conducted the following three case studies to illustrate the performance of the proposed algorithm.

#### 1. Case Study 1: On-Demand Air Transportation

In the first case study, based on the airspace configuration, we define a demand model that generates flight requests stochastically. At each vertiport, after the taking off of the previous aircraft, the time interval for next aircraft to take off is uniformly distributed between 1

**Table 1** Simulation results of MCTS algorithm averaged over five independent experiments

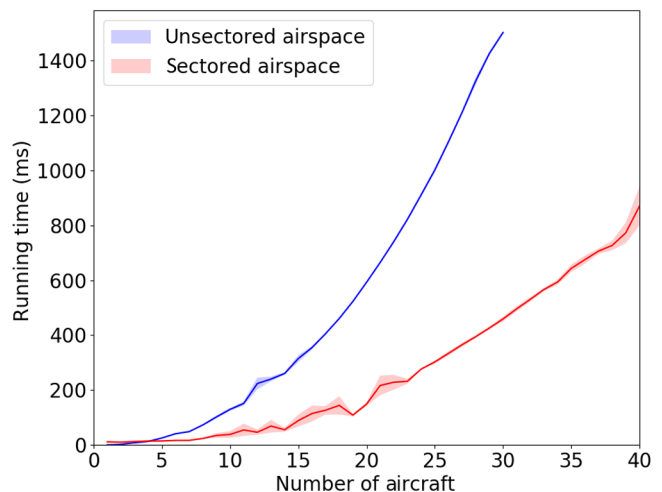
	Average LOS per flight hour	Average NMACs per flight hour
Sectored airspace	$(1.05 \pm 0.34) \times 10^{-2}$	$(1.45 \pm 2.90) \times 10^{-4}$
Unsectored airspace	$(1.31 \pm 0.22) \times 10^{-2}$	$(2.91 \pm 3.57) \times 10^{-4}$

and 2 min, and the newly generated flight request will choose a random vertiport as its destination. Then, the simulator will be kept running until 10,000 aircraft have been generated. During the running of this simulator, the number of LOS events and NMACs and the average computation time to make each decision will be recorded and compared. Here, the NMAC standard is defined to be 500 ft by the Aeronautical Information Manual (7-6-3) [130].

Then, we conducted five independent experiments, and there are 10,000 aircraft generated in total in each experiment. **The code implementation of this algorithm is available on GitHub** ([https://github.com/xuxiyang1993/Multi\\_MCTS\\_Guidance\\_Separation\\_Assurance](https://github.com/xuxiyang1993/Multi_MCTS_Guidance_Separation_Assurance)) [131] and a short video demonstration for this case study can be found on YouTube ([https://www.youtube.com/watch?v=2cbRUig4G\\_I](https://www.youtube.com/watch?v=2cbRUig4G_I)).

Table 1 shows the average number of LOS events and NMACs per flight hour and standard error from the simulation results over five independent experiments with and without sector configuration (referred to as sectored airspace and unsectored airspace in the following). From this table, we see that the LOS event happens around  $1 \times 10^{-2}$  per flight hour and NMAC happens around  $1 \times 10^{-4}$  per flight hour. This table shows that the proposed algorithm has promising performance for guidance and separation assurance, and the introduction of the airspace sector helps reduce the LOS/NMAC risk for aircraft. Note that the recorded number of NMACs during simulation is in the absence of a collision avoidance system, such as TCAS or ACAS X. A direction of future work would be to integrate our separation assurance model with a collision avoidance system as the final layer of protection. Furthermore, all results shown in this work have not been integrated with any strategic flight plan deconfliction, traffic flow management, or flow control. We expect better safety performance once integrating these components.

Figure 6 plots the computation time needed for all the aircraft to finish running the onboard algorithm to decide the joint actions. This shows that the computation time for both unsectored airspace and sectored airspace are growing with the increase of the number of aircraft, and **the sectored airspace greatly reduces the computation time to real-time levels. For the 30 aircraft cases, the algorithm with airspace sectorization only takes less than 500 ms to issue the actions for all the aircraft.** Note, here, we record the longest computation time among the seven sectors. Because most of the time the aircraft are not evenly distributed in the seven sectors, the computation time does not



**Fig. 6** Computation time with the increasing number of aircraft.



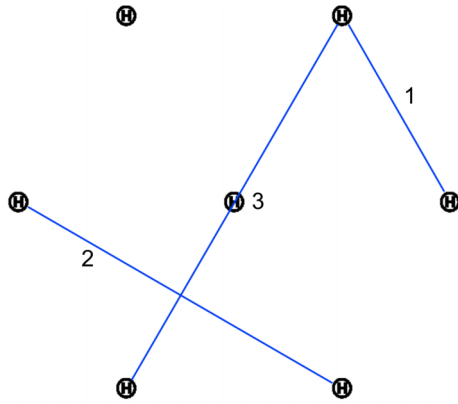


Fig. 7 Three different routes of the vertiport setting in the case study 1.

achieve seven times faster and the variance among five independent experiments is larger than that in unsectored airspace.

Note, here, that the MCTS algorithm is a statistical anytime algorithm [99]. This means the algorithm can stop running at any time by returning the current best action, and longer computation time generally leads to better action advisory. This is beneficial for the case when an aircraft needs a maximum acceptable run time.

Although the introduction of airspace sectorization helps increase the safety level and reduces the computational time of the proposed algorithm, it also requires metering the aircraft to the exit gate of each sector, which will force the vehicles to deviate from their preferred optimal route and result in suboptimal trajectories. Thus, in the numerical experiment, we also studied the en route flight time in sectorized airspace and unsectorized airspace. Here, we divided the route option into three categories based on the distance between the start vertiport and target vertiport. Recall in Fig. 4 that we labeled the vertiports with different ID numbers. In this vertiport map, route 1 is the shortest (e.g., from vertiport 1 to vertiport 2), route 2 has the medium length (e.g., from vertiport 3 to vertiport 7), and route 3 has the longest path (e.g., from vertiport 3 to vertiport 6). The three different routes are plotted in Fig. 7. Table 2 shows the en route flight time of sectorized airspace and unsectorized airspace for different route options assuming there is no intruder aircraft. From this table, we can see for routes 1, 2, and 3 that the airspace sectorization increases the flight time by 6, 20, and 7 s, respectively, which is acceptable in trade of safety level and computation time. For routes 1 and 3, the difference is small because the deviated trajectory in sectorized airspace is close to the preferred optimal trajectory in unsectorized airspace.

In Fig. 8, we recorded the number of en route aircraft at each time step and plotted the resulting histogram. In this plot,  $x$  axis is the total number of en route aircraft and  $y$  axis denotes their frequency. From this figure, we can see most of the time, there are 25 ~ 40 aircraft flying in the air, which is approximately 22.55 ~ 36.08 aircraft per 10,000 km<sup>2</sup>.

## 2. Case Study 2: eVTOL with Different Priorities

In the second case study, we further analyze the en route flight time for the aircraft with priority, because in some cases, we may have an emergency flight request, which needs to reach its destination in the shortest possible time (e.g., a law enforcement eVTOL or an ambulance eVTOL). This case can also happen when an aircraft flying in the air has a fault in its electric propulsion system. To let these emergent aircraft arrive goal in a shorter time, in this experiment, we divide all the aircraft into two categories: each aircraft is classified

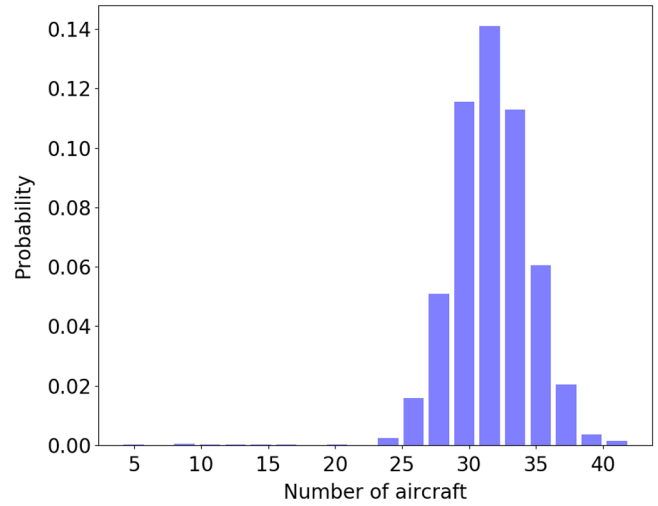


Fig. 8 Histogram for the total number of en route aircraft.

into high/low priority, which is used to denote the emergency level for an aircraft. When an aircraft takes off, the priority will be randomly assigned to the aircraft in the numerical experiment. In real-world applications, an eVTOL will be identified and approved with different priorities depending on its trip purpose.

When the algorithm makes decisions for the aircraft, it will first generate actions for all the aircraft with high priority, by only considering the high-priority aircraft information. Then, it will next generate actions for all the remaining aircraft, given the actions made for the high-priority aircraft.

Specifically, suppose in the current sector we have  $n$  aircraft, out of which  $p$  aircraft are with high priority. Then, the algorithm will first initialize the actions for these  $p$  aircraft:

$$\mathbf{a}_{\text{high}} = \{a_1, a_2, \dots, a_p\} \quad (15)$$

where  $a_i$  is the action for aircraft  $i$ , and  $a_1 = a_2 = \dots = a_p = 0$  deg/s. The algorithm will generate actions based on the high-priority aircraft information  $[s_1, \dots, s_p]$ , where  $s_i$  denotes the state information for aircraft  $i$ . After generating the optimal actions for these  $p$  high-priority aircraft, we get

$$\mathbf{a}_{\text{high}}^* = \{a_1^*, a_2^*, \dots, a_p^*\} \quad (16)$$

The algorithm will then next initialize the joint action for all the remaining aircraft as

$$\mathbf{a}_{\text{low}} = \{a_1^*, \dots, a_p^*, a_{p+1}, \dots, a_n\} \quad (17)$$

where  $a_{p+1} = \dots = a_n = 0$  deg/s. Then, the algorithm will generate actions for these aircraft with all of the aircraft information  $[s_1, \dots, s_n]$ .

In this experiment, we compare the average en route flight time of aircraft with different priorities for the three different routes, which is recorded in Table 3. In this table, we also added the optimal en route flight time where there is no intruder aircraft, from which we can see that the flight time of high-priority aircraft is closer to the optimal path flight time compared with low-priority aircraft, and high-priority

Table 2 Flight time (en route air time) in seconds for sectorized airspace and unsectorized airspace

Route	1	2	3
Sectorized airspace	340.60 ± 7.91	606.56 ± 10.58	685.71 ± 10.81
Unsectorized airspace	334.46 ± 7.48	586.43 ± 9.88	678.81 ± 10.75

Table 3 Flight time (en route air time) in seconds for high-priority aircraft and low-priority aircraft

Route	1	2	3
Clear path	340.60 ± 7.91	606.56 ± 10.58	685.71 ± 10.81
High priority	344.93 ± 0.32	619.98 ± 0.70	703.14 ± 1.31
Low priority	358.80 ± 0.83	653.38 ± 0.49	746.51 ± 2.32

aircraft can save 14, 34, and 43 s en route time for each route compared with low-priority aircraft.

### 3. Case Study 3: Roundabout Stress Test

In this case study, the proposed algorithm is evaluated against a stress-test set of multithreat scenarios randomly generated from an encounter model, similar to the experiments presented in [69]. This case study is helpful because a large part of current commercial interest surrounding UAM stems from the potential to accommodate a large number of eVTOL aircraft, and it is estimated that there will be 23,000 aircraft flying major routes within the UAM network by 2035 [5].

In each encounter scenario, the number of aircraft in the multithreat encounters is ranging from 10 to 20, distributing uniformly on an annulus. Specifically, the annulus had inner and outer radii of 10 and 15 km, and if a new aircraft added is closer to other aircraft than 2 km, we resample the new aircraft position to avoid initializing aircraft already in the LOS state. The goal position of each aircraft is set to be the symmetric point in the annulus with respect to this aircraft's position, and so the headings of each aircraft are initialized to point straight toward the annulus center to ensure that all aircraft would have potential conflicts. Figure 9 shows an example stress-test scenario with 10 uniformly initialized aircraft, where each aircraft and its destination are connected through a dashed line.

Figure 10 illustrates the performance of the proposed computational guidance algorithm in the stress-test scenarios as the number of aircraft increases, where each point denotes the average result in 1000 independent encounter scenarios. It plots the probability that an aircraft has LOS/NMAC with another aircraft in one scenario. From Fig. 10a, we can see for the MCTS algorithm that the LOS event probability is less than 1% and the NMAC probability is less than 0.1% for each aircraft. We also show the result of the baseline, where no actions are taken for all of the aircraft and they are flying straight toward their respective goals, and plot the average result over 1000 independent experiments. To ensure the aircraft can arrive at their respective goals, we remove the uncertainty for the change of heading angle and only keep the uncertainty of speed. From Fig. 10b, we can see that, when no actions are taken, over 90% of the aircraft will have at least one LOS event and over 40% of the aircraft will have NMAC with other aircraft in each stress-test encounter scenario. The comparison of the two plots in Fig. 10 shows the promising performance of the proposed computational guidance algorithm even with high-density air traffic.

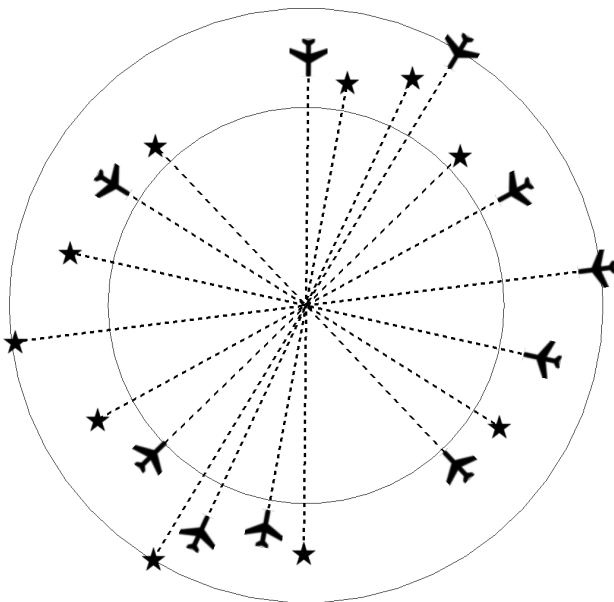
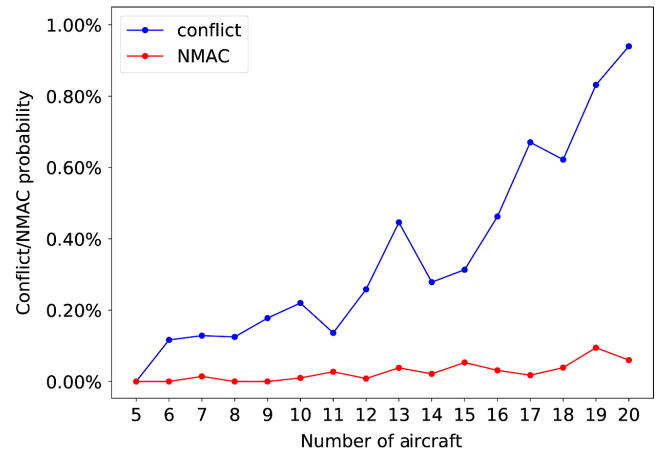
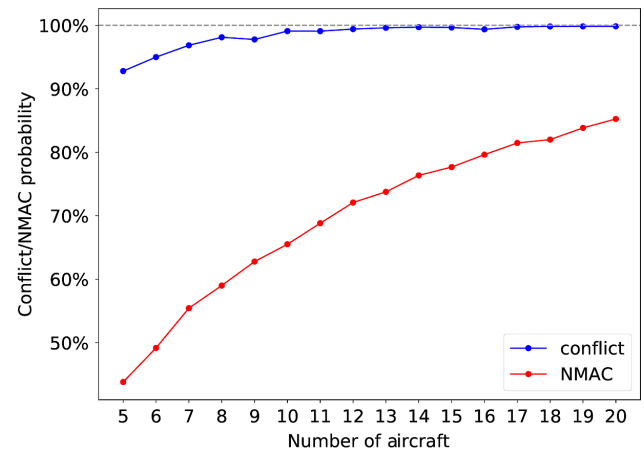


Fig. 9 One randomly generated stress-test scenario.



a) MCTS algorithm



b) Baseline

Fig. 10 LOS event/NMAC probability as the number of aircraft increases.

## VI. Conclusions

A message-passing decentralized computational guidance algorithm with a separation assurance capability for multiple cooperative aircraft in UAM is proposed in this paper. The problem is formulated as an MMDP, and then solved by MCTS algorithm that can run onboard. A message-passing-based coordination mechanism was designed to manage multiple cooperative aircraft. The airspace sectorization is introduced to help achieve better scalability and safety. Numerical experiments over three case studies show that this proposed algorithm has promising performance to help the aircraft reach their destinations and avoid potential LOS events among them even for the high-density air traffic case. The contributions of this paper include 1) exploring the feasibility of the free-flight concept of operations for UAM, 2) designing the multi-agent computational guidance algorithm with a separation assurance capability, and 3) proposing airspace sectorization in UAM.

The proposed concept of operations and algorithm provides a potential solution for distributed separation assurance to enable safe, efficient, and scalable flight operations in on-demand urban air transportation with high-density air traffic. This framework might be a potential solution for certain airspace, such as rural or suburban areas. However, the proposed framework is still in the exploratory phase with some limitations. To make the proposed algorithm more practical in real-world applications, the future work should include 1) expanding the action space to altitude changes and speed changes; 2) adapting the proposed algorithm in restricted airspace or structured airspace; 3) incorporating a high-fidelity aircraft dynamics model with more realistic mission profile; and 4) integrating other layers of protection into this framework, such as strategic flight planning, flow control, and collision avoidance systems.

## Acknowledgment

This research is partially supported by the National Science Foundation grant number 1565979.

## References

- [1] Gipson, L., "NASA Embraces Urban Air Mobility, Calls for Market Study," 2017, <https://www.nasa.gov/aero/nasa-embraces-urban-air-mobility> [retrieved 15 Feb. 2020].
- [2] Thippavong, D. P., Apaza, R., Barmore, B., Battiste, V., Burian, B., Dao, Q., Feary, M., Go, S., Goodrich, K. H., Homola, J., et al., "Urban Air Mobility Airspace Integration Concepts and Considerations," 2018 *Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2018-3676, June 2018. <https://doi.org/10.2514/6.2018-3676>
- [3] Moore, M., "Uber Elevate: eVTOL Urban Mobility," *Rotorcraft Business & Technology Summit*, 2017.
- [4] Anon., "Future of Urban Mobility," 2017, <https://www.airbus.com/newsroom/news/en/2016/12/My-Kind-Of-Flyover.html> [retrieved 15 Feb. 2020].
- [5] Anon., "The Future of Vertical Mobility," March 2018, [https://www.porsche-consulting.com/fileadmin/docs/04\\_Medien/Publikationen/TT1371\\_The\\_Future\\_of\\_Vertical\\_Mobility/The\\_Future\\_of\\_Vertical\\_Mobility\\_A\\_Porsche\\_Consulting\\_study\\_C\\_2018.pdf](https://www.porsche-consulting.com/fileadmin/docs/04_Medien/Publikationen/TT1371_The_Future_of_Vertical_Mobility/The_Future_of_Vertical_Mobility_A_Porsche_Consulting_study_C_2018.pdf) [retrieved 15 Feb. 2020].
- [6] Mueller, E. R., Kopardekar, P. H., and Goodrich, K. H., "Enabling Airspace Integration for High-Density On-Demand Mobility Operations," 17th *AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-3086, June 2017. <https://doi.org/10.2514/6.2017-3086>
- [7] Anon., "NextGen Implementation Plan 2016," U.S. Department of Transportation, 2016, [https://www.faa.gov/nextgen/media/nextgen\\_implementation\\_plan-2016.pdf](https://www.faa.gov/nextgen/media/nextgen_implementation_plan-2016.pdf) [retrieved 25 April 2020].
- [8] Gawdiak, Y., Carr, G., and Hasan, S., "JPDO Case Study of NextGen High Density Operations," 9th *AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, AIAA Paper 2009-6918, June 2009. <https://doi.org/10.2514/6.2009-6918>
- [9] Timar, S., Hunter, G., and Post, J., "Assessing the Benefits of NextGen Performance Based Navigation (PBN)," *Air Traffic Control Quarterly*, Vol. 21, No. 3, June 2013, pp. 211–232.
- [10] Moore, M. D., and Goodrich, K. H., "High Speed Mobility Through On-Demand Aviation," 2013 *Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2013-4373, Aug. 2013. <https://doi.org/10.2514/6.2013-4373>
- [11] Gawdiak, Y., Holmes, B., Sawhill, B., Herriot, J., Ballard, D., Creedon, J., Eckhause, J., Long, D., Hemm, R., Murphy, C., et al., "Air Transportation Strategic Trade Space Modeling and Assessment Through Analysis of On-Demand Air Mobility with Electric Aircraft," 12th *AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2012-5594, Sept. 2012.
- [12] Zhu, G., and Wei, P., "Pre-Departure Planning for Urban Air Mobility Flights with Dynamic Airspace Reservation," *AIAA Aviation 2019 Forum*, AIAA Paper 2019-3519, June 2019.
- [13] Force, R. T., "Final Report of RTCA Task Force 3: Free Flight Implementation," Tech. Rept., Radio Technical Commission for Aeronautics, 1995.
- [14] Hoekstra, J. M., van Gent, R. N., and Ruigrok, R. C., "Designing for Safety: The 'Free Flight' Air Traffic Management Concept," *Reliability Engineering & System Safety*, Vol. 75, No. 2, 2002, pp. 215–232. [https://doi.org/10.1016/S0951-8320\(01\)00096-5](https://doi.org/10.1016/S0951-8320(01)00096-5)
- [15] Bilimoria, K. D., Grabbe, S. R., Sheth, K. S., and Lee, H. Q., "Performance Evaluation of Airborne Separation Assurance for Free Flight," *Air Traffic Control Quarterly*, Vol. 11, No. 2, 2003, pp. 85–102. <https://doi.org/10.2514/atcq.11.2.85>
- [16] Consiglio, M., Hoadley, S., Wing, D., and Baxley, B., "Safety Performance of Airborne Separation: Preliminary Baseline Testing," 7th *AIAA ATIO Conf, 2nd CEIAT Int'l Conf on Innov and Integr in Aero Sciences, 17th LTA Systems Tech Conf; followed by 2nd TEOS Forum*, AIAA Paper 2007-7739, Jan. 2007. <https://doi.org/10.2514/6.2007-7739>
- [17] Blom, H. A., and Bakker, G., "Safety Evaluation of Advanced Self-Separation Under Very High En Route Traffic Demand," *Journal of Aerospace Information Systems*, Vol. 12, No. 6, 2015, pp. 413–427. <https://doi.org/10.2514/1.1010243>
- [18] Clari, M. S. V., Ruigrok, R. C., Hoekstra, J. M., and Visser, H. G., "Cost-Benefit Study of Free Flight with Airborne Separation Assurance," *Air Traffic Control Quarterly*, Vol. 9, No. 4, 2001, pp. 287–309. <https://doi.org/10.2514/atcq.9.4.287>
- [19] Wing, D. J., Richard, J. A., Jacqueline, A. D., Brain, M. L., Bryan, E. B., and Donald, M., "Airborne Use of Traffic Intent Information in a Distributed Air-Ground Traffic Management Concept: Experiment Design and Preliminary Results," NASA Langley Research Center, Nov. 2001.
- [20] Battiste, V., Johnson, W., Kopardekar, P., Lozito, S., Mogford, R., and Palmer, E., "Distributed Air/Ground Traffic Management—Technology and Concept Demonstration Report," *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum*, AIAA Paper 2002-5825, June 2002. <https://doi.org/10.2514/6.2002-5825>
- [21] Barhydt, R., Kopardekar, P., Battiste, V., Doble, N., Johnson, W., Lee, P., Prevot, T., and Smith, N., "Joint NASA Ames/Langley Experimental Evaluation of Integrated Air/Ground Operations for En Route Free Maneuvering," 6th *USA/Europe Seminar on Air Traffic Management Research and Development*, June 2005.
- [22] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 4, 1998, pp. 509–521. <https://doi.org/10.1109/9.664154>
- [23] Kahne, S., and Frowlow, I., "Air Traffic Management: Evolution with Technology," *IEEE Control Systems*, Vol. 16, No. 4, 1996, pp. 12–21. <https://doi.org/10.1109/37.526911>
- [24] Kochenderfer, M. J., Holland, J. E., and Chryssanthacopoulos, J. P., "Next-Generation Airborne Collision Avoidance System," Tech. Rept., Massachusetts Inst. of Technology Lincoln Lab., Lexington, MA, 2012.
- [25] Kuchar, J. K., and Yang, L. C., "A Review of Conflict Detection and Resolution Modeling Methods," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 1, No. 4, 2000, pp. 179–189. <https://doi.org/10.1109/6979.898217>
- [26] Netjasov, F., and Janic, M., "A Review of Research on Risk and Safety Modelling in Civil Aviation," *Journal of Air Transport Management*, Vol. 14, No. 4, 2008, pp. 213–220. <https://doi.org/10.1016/j.jairtraman.2008.04.008>
- [27] Mitici, M., and Blom, H. A. P., "Mathematical Models for Air Traffic Conflict and Collision Probability Estimation," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 20, No. 3, 2019, pp. 1052–1068. <https://doi.org/10.1109/TITS.2018.2839344>
- [28] Kochenderfer, M. J., and Chryssanthacopoulos, J., "Robust Airborne Collision Avoidance Through Dynamic Programming," Massachusetts Inst. of Technology Lincoln Lab. Project Rept. ATC-371, Lexington, MA, 2011.
- [29] Karr, D., Vivona, R., Roscoe, D., Depascale, S., and Wing, D., "Autonomous Operations Planner: A Flexible Platform for Research in Flight-Deck Support for Airborne Self-Separation," 12th *AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, AIAA Paper 2012-5417, Sept. 2012. <https://doi.org/10.2514/6.2012-5417>
- [30] Prevot, T., Rios, J., Kopardekar, P., Robinson, J. E., III, Johnson, M., and Jung, J., "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," 16th *AIAA Aviation Technology, Integration, and Operations Conference*, June 2016, pp. 32–92.
- [31] Consiglio, M., Muñoz, C., Hagen, G., Narkawicz, A., and Balachandran, S., "ICAROUS: Integrated Configurable Algorithms for Reliable Operations of Unmanned Systems," 2016 *IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, New York, 2016, pp. 1–5.
- [32] Muñoz, C., Narkawicz, A., Hagen, G., Upchurch, J., Dutle, A., Consiglio, M., and Chamberlain, J., "DAIDALUS: Detect and Avoid Alerting Logic for Unmanned Systems," 34th *Digital Avionics Systems Conference (DASC)*, IEEE, New York, 2015, pp. 5A1-1–5A1-12. <https://doi.org/10.1109/DASC.2015.7311421>
- [33] Jung, J., D'Souza, S. N., Johnson, M. A., Ishihara, A. K., Modi, H. C., Nikaido, B., and Hasseeb, H., "Applying Required Navigation Performance Concept for Traffic Management of Small Unmanned Aircraft Systems," 30th *Congress of the International Council of the Aeronautical Sciences*, Sept. 2016.
- [34] Schouwenaars, T., How, J., and Feron, E., "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2004-5141, June 2004. <https://doi.org/10.2514/6.2004-5141>



- [35] Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D., *Introduction to Autonomous Mobile Robots*, MIT Press, Cambridge, MA, 2011, pp. 257–290.
- [36] Frazzoli, E., Mao, Z.-H., Oh, J.-H., and Feron, E., “Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 24, No. 1, 2001, pp. 79–86.  
<https://doi.org/10.2514/2.4678>
- [37] Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., and Samad, T., “Dynamic Optimization Strategies for Three-Dimensional Conflict Resolution of Multiple Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 4, 2004, pp. 586–594.  
<https://doi.org/10.2514/1.11168>
- [38] Enright, P. J., and Conway, B. A., “Discrete Approximations to Optimal Trajectories Using Direct Transcription and Nonlinear Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 994–1002.  
<https://doi.org/10.2514/3.20934>
- [39] Schouwenaars, T., De Moor, B., Feron, E., and How, J., “Mixed Integer Programming for Multi-Vehicle Path Planning,” *2001 European Control Conference (ECC)*, IEEE, New York, 2001, pp. 2603–2608.  
<https://doi.org/10.23919/ECC.2001.7076321>
- [40] Richards, A., and How, J. P., “Aircraft Trajectory Planning with Collision Avoidance Using Mixed Integer Linear Programming,” *Proceedings of the 2002 American Control Conference*, Vol. 3, IEEE, New York, 2002, pp. 1936–1941.  
<https://doi.org/10.1109/ACC.2002.1023918>
- [41] Pallottino, L., Feron, E. M., and Bicchi, A., “Conflict Resolution Problems for Air Traffic Management Systems Solved with Mixed Integer Programming,” *IEEE Transactions on Intelligent Transportation Systems*, Vol. 3, No. 1, 2002, pp. 3–11.  
<https://doi.org/10.1109/6979.994791>
- [42] Vela, A., Solak, S., Singhose, W., and Clarke, J.-P., “A Mixed Integer Program for Flight-Level Assignment and Speed Control for Conflict Resolution,” *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 Held Jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009*, IEEE, New York, 2009, pp. 5219–5226.  
<https://doi.org/10.1109/CDC.2009.5400520>
- [43] Mellinger, D., Kushleyev, A., and Kumar, V., “Mixed-Integer Quadratic Program Trajectory Generation for Heterogeneous Quadrotor Teams,” *2012 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2012, pp. 477–483.  
<https://doi.org/10.1109/ICRA.2012.6225009>
- [44] Augugliaro, F., Schoellig, A. P., and D’Andrea, R., “Generation of Collision-Free Trajectories for a Quadcopter Fleet: A Sequential Convex Programming Approach,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, New York, 2012, pp. 1917–1922.  
<https://doi.org/10.1109/iros.2012.6385823>
- [45] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., “Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming,” *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.  
<https://doi.org/10.2514/1.G000218>
- [46] Delahaye, D., Peyronne, C., Mongeau, M., and Puechmorel, S., “Aircraft Conflict Resolution by Genetic Algorithm and B-Spline Approximation,” *EIWAC 2010, 2nd ENRI International Workshop on ATM/CNS*, Tokyo, Japan, Nov. 2010, pp. 71–78.
- [47] Cobano, J. A., Conde, R., Alejo, D., and Ollero, A., “Path Planning Based on Genetic Algorithms and the Monte-Carlo Method to Avoid Aerial Vehicle Collisions Under Uncertainties,” *2011 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2011, pp. 4429–4434.  
<https://doi.org/10.1109/ICRA.2011.5980246>
- [48] Pontani, M., and Conway, B. A., “Particle Swarm Optimization Applied to Space Trajectories,” *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 5, 2010, pp. 1429–1441.  
<https://doi.org/10.2514/1.48475>
- [49] Hoffmann, G., Rajnarayan, D. G., Waslander, S. L., Dostal, D., Jang, J. S., and Tomlin, C. J., “The Stanford Testbed of Autonomous Rotorcraft for Multi Agent Control (STARMAC),” *23rd Digital Avionics Systems Conference*, Vol. 2, IEEE, New York, 2004, p. 12-E.
- [50] Howlet, J. K., Schuelein, G., and Mansur, M. H., “A Practical Approach to Obstacle Field Route Planning for Unmanned Rotorcraft,” *American Helicopter Society 60th Annual Forum*, June 2004.
- [51] Canny, J., *The Complexity of Robot Motion Planning*, MIT Press, Cambridge, MA, 1988.
- [52] Kavradi, L., Svestka, P., and Overmars, M. H., “Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces,” *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1994, pp. 566–580.
- [53] LaValle, S. M., “Rapidly-Exploring Random Trees: A New Tool for Path Planning,” Iowa State Univ., TR-98-11, Ames, IA, Oct. 1998.
- [54] Karaman, S., and Frazzoli, E., “Sampling-Based Algorithms for Optimal Motion Planning,” *International Journal of Robotics Research*, Vol. 30, No. 7, 2011, pp. 846–894.
- [55] Čáp, M., Novák, P., Vokrněk, J., and Pěchouček, M., “Multi-Agent RRT: Sampling-Based Cooperative Pathfinding,” *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, International Foundation for Autonomous Agents and Multiagent Systems, Saint Paul, Minnesota, 2013, pp. 1263–1264.
- [56] Pallottino, L., Scordio, V. G., Frazzoli, E., and Bicchi, A., “Probabilistic Verification of a Decentralized Policy for Conflict Resolution in Multi-Agent Systems,” *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, IEEE, New York, 2006, pp. 2448–2453.  
<https://doi.org/10.1109/ROBOT.2006.1642069>
- [57] Wollkind, S., Valasek, J., and Ioerger, T., “Automated Conflict Resolution for Air Traffic Management Using Cooperative Multiagent Negotiation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2004-4992, June 2004.
- [58] Purwin, O., D’Andrea, R., and Lee, J.-W., “Theory and Implementation of Path Planning by Negotiation for Decentralized Agents,” *Robotics and Autonomous Systems*, Vol. 56, No. 5, 2008, pp. 422–436.  
<https://doi.org/10.1016/j.robot.2007.09.020>
- [59] Desraj, V. R., and How, J. P., “Decentralized Path Planning for Multi-Agent Teams in Complex Environments Using Rapidly-Exploring Random Trees,” *2011 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2011, pp. 4956–4961.  
<https://doi.org/10.1109/ICRA.2011.5980392>
- [60] Inalhan, G., Stipanovic, D. M., and Tomlin, C. J., “Decentralized Optimization, with Application to Multiple Aircraft Coordination,” *Proceedings of the 41st IEEE Conference on Decision and Control, 2002*, Vol. 1, IEEE, New York, 2002, pp. 1147–1155.  
<https://doi.org/10.1109/CDC.2002.1184667>
- [61] Richards, A., and How, J., “Decentralized Model Predictive Control of Cooperating UAVs,” *43rd IEEE Conference on Decision and Control (CDC)*, Vol. 4, IEEE, New York, 2004, pp. 4286–4291.  
<https://doi.org/10.1109/CDC.2004.1429425>
- [62] Shim, D. H., and Sastry, S., “An Evasive Maneuvering Algorithm for UAVs in See-and-Avoid Situations,” *2007 American Control Conference*, IEEE, New York, 2007, pp. 3886–3891.  
<https://doi.org/10.1109/ACC.2007.4283147>
- [63] Shim, D. H., Kim, H. J., and Sastry, S., “Decentralized Nonlinear Model Predictive Control of Multiple Flying Robots,” *42nd IEEE Conference on Decision and Control*, Vol. 4, IEEE, New York, 2003, pp. 3621–3626.  
<https://doi.org/10.1109/CDC.2003.1271710>
- [64] Khatib, O., “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,” *Autonomous Robot Vehicles*, edited by I. J. Cox, and G. T. Wilfong, Springer, New York, 1990, pp. 396–404.  
[10.1007/978-1-4613-8997-2\\_29](https://doi.org/10.1007/978-1-4613-8997-2_29)
- [65] Koditschek, D. E., and Rimon, E., “Robot Navigation Functions on Manifolds with Boundary,” *Advances in Applied Mathematics*, Vol. 11, No. 4, 1990, pp. 412–442.
- [66] Rimon, E., and Koditschek, D. E., “Exact Robot Navigation Using Cost Functions: The Case of Distinct Spherical Boundaries in  $E/\sup n$ ,” *1988 IEEE International Conference on Robotics and Automation*, IEEE, New York, 1988, pp. 1791–1796.
- [67] Connolly, C. I., Burns, J. B., and Weiss, R., “Path Planning Using Laplace’s Equation,” *1990 IEEE International Conference on Robotics and Automation*, IEEE, New York, 1990, pp. 2102–2106.  
<https://doi.org/10.1109/ROBOT.1990.126315>
- [68] Kahn, G., Zhang, T., Levine, S., and Abbeel, P., “Plato: Policy Learning Using Adaptive Trajectory Optimization,” *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, New York, 2017, pp. 3342–3349.  
<https://doi.org/10.1109/ICRA.2017.7989379>
- [69] Ong, H. Y., and Kochenderfer, M. J., “Markov Decision Process-Based Distributed Conflict Resolution for Drone Air Traffic Management,” *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1, 2017, pp. 69–80.  
<https://doi.org/10.2514/1.G001822>
- [70] Chen, Y. F., Liu, M., Everett, M., and How, J. P., “Decentralized Non-Communicating Multiagent Collision Avoidance with Deep Reinforcement Learning,” *2017 IEEE International Conference on*

- Robotics and Automation (ICRA)*, IEEE, New York, 2017, pp. 285–292. <https://doi.org/10.1109/ICRA.2017.7989037>
- [71] Li, S., Egorov, M., and Kochenderfer, M., “Optimizing Collision Avoidance in Dense Airspace Using Deep Reinforcement Learning,” *13th USA/Europe Air Traffic Management Research and Development Seminar*, June 2019.
- [72] Brittain, M., and Wei, P., “Autonomous Separation Assurance in a High-Density En Route Sector: A Deep Multi-Agent Reinforcement Learning Approach,” *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, IEEE, New York, 2019, pp. 3256–3262. <https://doi.org/10.1109/ITSC.2019.8917217>
- [73] Yang, X., and Wei, P., “Autonomous On-Demand Free Flight Operations in Urban Air Mobility Using Monte Carlo Tree Search,” *8th International Conference on Research in Air Transportation (ICRAT)*, ICRAT, 2018.
- [74] Wolf, T. B., and Kochenderfer, M. J., “Aircraft Collision Avoidance Using Monte Carlo Real-Time Belief Space Search,” *Journal of Intelligent & Robotic Systems*, Vol. 64, No. 2, 2011, pp. 277–298. <https://doi.org/10.1007/s10846-010-9532-6>
- [75] Han, S.-C., Bang, H., and Yoo, C.-S., “Proportional Navigation-Based Collision Avoidance for UAVs,” *International Journal of Control, Automation and Systems*, Vol. 7, No. 4, 2009, pp. 553–565. <https://doi.org/10.1007/s12555-009-0407-1>
- [76] Park, J.-W., Oh, H.-D., and Tahk, M.-J., “UAV Collision Avoidance Based on Geometric Approach,” *2008 SICE Annual Conference*, IEEE, New York, 2008, pp. 2122–2126. <https://doi.org/10.1109/SICE.2008.4655013>
- [77] Krozel, J., Peters, M., and Bilimoria, K., “A Decentralized Control Strategy for Distributed Air/Ground Traffic Separation,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, AIAA Paper 2000-4062, Aug. 2000. <https://doi.org/10.2514/6.2000-4062>
- [78] van den Berg, J., Guy, S. J., Lin, M., and Manocha, D., “Reciprocal n-Body Collision Avoidance,” *Robotics Research*, Springer, Berlin, 2011, pp. 3–19. [https://doi.org/10.1007/978-3-642-19457-3\\_1](https://doi.org/10.1007/978-3-642-19457-3_1)
- [79] Yang, X., Deng, L., Liu, J., Wei, P., and Li, H., “Multi-Agent Autonomous Operations in Urban Air Mobility with Communication Constraints,” *AIAA SciTech 2020 Forum*, AIAA Paper 2020-1839, Jan. 2020. <https://doi.org/10.2514/6.2020-1839>
- [80] Temizer, S., Kochenderfer, M., Kaelbling, L., Lozano-Pérez, T., and Kuchar, J., “Collision Avoidance for Unmanned Aircraft Using Markov Decision Processes,” *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2010-8040, Aug. 2010.
- [81] Bertram, J., and Wei, P., “Distributed Computational Guidance for High-Density Urban Air Mobility with Cooperative and Non-Cooperative Collision Avoidance,” *AIAA SciTech 2020 Forum*, AIAA Paper 2020-1371, Jan. 2020. <https://doi.org/10.2514/6.2020-1371>
- [82] Anon., “Control Sector,” 2019, <https://www.faraim.org/aim/aim-4-03-14-618.html> [retrieved 15 Feb. 2020].
- [83] Anon., “Introduction to Airspace,” 2019, <https://www.nats.aero/ae-home/introduction-to-airspace/> [retrieved 15 Feb. 2020].
- [84] Anon., “Loss of Separation at Sector Boundaries,” 2019, [https://www.skybrary.aero/index.php/Loss\\_of\\_Separation\\_at\\_Sector\\_Boundaries#](https://www.skybrary.aero/index.php/Loss_of_Separation_at_Sector_Boundaries#) [retrieved 15 Feb. 2020].
- [85] Pradeep, P., and Wei, P., “Energy Optimal Speed Profile for Arrival of Tandem Tilt-Wing eVTOL Aircraft with RTA Constraint,” *2018 IEEE CSAA Guidance, Navigation and Control Conference*, AIAA Paper 2018-2008, Aug. 2018. <https://doi.org/10.2514/6.2018-2008>
- [86] Anon., “Vahana,” 2018, <https://www.airbus.com/innovation/urban-air-mobility/vehicle-demonstrators/vahana.html> [retrieved 15 Feb. 2020].
- [87] Pradeep, P., and Wei, P., “Heuristic Approach for Arrival Sequencing and Scheduling for eVTOL Aircraft in On-Demand Urban Air Mobility,” *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, New York, 2018, pp. 1–7.
- [88] Kleinbekman, I. C., Mitici, M. A., and Wei, P., “eVTOL Arrival Sequencing and Scheduling for On-Demand Urban Air Mobility,” *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, IEEE, New York, 2018, pp. 1–7.
- [89] Bertram, J., and Wei, P., “An Efficient Algorithm for Self-Organized Terminal Arrival in Urban Air Mobility,” *AIAA SciTech 2020 Forum*, AIAA Paper 2020-0660, Jan. 2020. <https://doi.org/10.2514/6.2020-0660>
- [90] Stoschek, A., “Exploring Sense-and-Avoid Systems for Autonomous Vehicles,” Dec. 2017, <https://acubed.airbus.com/blog/vahana/exploring-sense-and-avoid-systems-for-autonomous-vehicles/> [retrieved 15 Feb. 2020].
- [91] Bellman, R., “A Markovian Decision Process,” *Indiana University Mathematics Journal*, Vol. 6, Jan. 1957, pp. 679–684.
- [92] Howard, R. A., *Dynamic Programming and Markov Processes*, American Association for the Advancement of Science, 1964, p. 667. <https://doi.org/10.1126/science.132.3428.667>
- [93] White, D. J., “A Survey of Applications of Markov Decision Processes,” *Journal of the Operational Research Society*, Vol. 44, No. 11, 1993, pp. 1073–1096. <https://doi.org/10.1057/jors.1993.181>
- [94] Feinberg, E. A., and Shwartz, A., *Handbook of Markov Decision Processes: Methods and Applications*, Vol. 40, Springer Science+Business Media, 2012, pp. 461–539. <https://doi.org/10.1007/978-1-4615-0805-2>
- [95] Koenig, S., and Simmons, R., “Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models,” *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, edited by D. Kortenkamp, R. Bonasso, and R. Murphy, MIT Press, Jan. 1998, pp. 91–122.
- [96] Thrun, S., “Probabilistic Robotics,” *Communications of the ACM*, Vol. 45, No. 3, 2002, pp. 52–57. <https://doi.org/10.1017/S0269888906210993>
- [97] Mariton, M., *Jump Linear Systems in Automatic Control*, M. Dekker, New York, 1990.
- [98] Coulom, R., “Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search,” *International Conference on Computers and Games*, Springer, Berlin, 2006, pp. 72–83. [https://doi.org/10.1007/978-3-540-75538-8\\_7](https://doi.org/10.1007/978-3-540-75538-8_7)
- [99] Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S., “A Survey of Monte Carlo Tree Search Methods,” *IEEE Transactions on Computational Intelligence and AI in Games*, Vol. 4, No. 1, 2012, pp. 1–43. <https://doi.org/10.1109/TCIAIG.2012.2186810>
- [100] Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., “Mastering the Game of Go with Deep Neural Networks and Tree Search,” *Nature*, Vol. 529, No. 7587, 2016, p. 484. <https://doi.org/10.1038/nature16961>
- [101] Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al., “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm,” arXiv preprint arXiv:1712.01815, 2017.
- [102] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al., “Mastering the Game of Go Without Human Knowledge,” *Nature*, Vol. 550, No. 7676, 2017, p. 354. <https://doi.org/10.1038/nature24270>
- [103] Boutilier, C., “Planning, Learning and Coordination in Multiagent Decision Processes,” *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, Morgan Kaufmann, San Mateo, CA, 1996, pp. 195–210.
- [104] Boutilier, C., Dean, T., and Hanks, S., “Decision-Theoretic Planning: Structural Assumptions and Computational Leverage,” *Journal of Artificial Intelligence Research*, Vol. 11, July 1999, pp. 1–94.
- [105] Boutilier, C., “Sequential Optimality and Coordination in Multiagent Systems,” *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Vol. 1, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999, pp. 478–485.
- [106] Sigaud, O., and Buffet, O., *Markov Decision Processes in Artificial Intelligence*, John Wiley & Sons, Hoboken, NJ, 2013, pp. 280–280.
- [107] Busoniu, L., Babuška, R., and De Schutter, B., “Multi-Agent Reinforcement Learning: An Overview,” *Innovations in Multi-Agent Systems and Applications-1*, Vol. 310, Studies in Computational Intelligence, Springer, Berlin, 2010, pp. 183–221.
- [108] Nash, J. F., “Equilibrium Points in n-Person Games,” *Proceedings of the National Academy of Sciences*, Vol. 36, No. 1, 1950, pp. 48–49.
- [109] Kochenderfer, M. J., *Decision Making Under Uncertainty: Theory and Application*, MIT Press, Cambridge, MA, 2015, pp. 68–72.
- [110] Daskalakis, C., Goldberg, P. W., and Papadimitriou, C. H., “The Complexity of Computing a Nash Equilibrium,” *SIAM Journal on Computing*, Vol. 39, No. 1, 2009, pp. 195–259.
- [111] Camerer, C. F., *Behavioral Game Theory: Experiments in Strategic Interaction*, Princeton Univ. Press, Princeton, NJ, 2011, pp. 20–23.
- [112] Stahl, D. O., II, and Wilson, P. W., “Experimental Evidence on Players’ Models of Other Players,” *Journal of Economic Behavior & Organization*, Vol. 25, No. 3, 1994, pp. 309–327.

- [113] Stahl, D. O., and Wilson, P. W., "On Players' Models of Other Players: Theory and Experimental Evidence," *Games and Economic Behavior*, Vol. 10, No. 1, 1995, pp. 218–254.
- [114] Julian, K. D., Lopez, J., Brush, J. S., Owen, M. P., and Kochenderfer, M. J., "Policy Compression for Aircraft Collision Avoidance Systems," *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, IEEE, New York, 2016, pp. 1–10.  
<https://doi.org/10.1109/DASC.2016.7778091>
- [115] Julian, K. D., Kochenderfer, M. J., and Owen, M. P., "Deep Neural Network Compression for Aircraft Collision Avoidance Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 3, 2018, pp. 598–608.
- [116] Tesauro, G., and Galperin, G. R., "On-Line Policy Improvement Using Monte-Carlo Search," *Proceedings of the 9th International Conference on Neural Information Processing Systems*, MIT Press, Cambridge, MA, 1996, pp. 1068–1074.
- [117] Chaslot, G. M. J., Winands, M. H., Herik, H. J. V. D., Uiterwijk, J. W., and Bouzy, B., "Progressive Strategies for Monte-Carlo Tree Search," *New Mathematics and Natural Computation*, Vol. 4, No. 3, 2008, pp. 343–357.
- [118] Coulom, R., "Computing 'Lo Ratings' of Move Patterns in the Game of Go," *ICGA Journal*, Vol. 30, No. 4, 2007, pp. 198–208.
- [119] Wang, Y., Audibert, J.-Y., and Munos, R., "Algorithms for Infinitely Many-Armed Bandits," *Advances in Neural Information Processing Systems*, Curran Associates, Vancouver, British Columbia, Canada, 2009, pp. 1729–1736.
- [120] Couëtoux, A., Hoock, J.-B., Sokolovska, N., Teytaud, O., and Bonnard, N., "Continuous Upper Confidence Trees," *International Conference on Learning and Intelligent Optimization*, Springer, Berlin, 2011, pp. 433–445.
- [121] Julian, K. D., and Kochenderfer, M. J., "Neural Network Guidance for UAVs," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2017-1743, Jan. 2017.
- [122] Bosson, C., and Lauderdale, T. A., "Simulation Evaluations of an Autonomous Urban Air Mobility Network Management and Separation Service," *2018 Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2018-3365, June 2018.  
<https://doi.org/10.2514/6.2018-3365>
- [123] Cook, S. P., and Brooks, D., "A Quantitative Metric to Enable Unmanned Aircraft Systems to Remain Well Clear," *Air Traffic Control Quarterly*, Vol. 23, Nos. 2–3, 2015, pp. 137–156.  
<https://doi.org/10.2514/atcq.23.2-3.137>
- [124] Kocsis, L., Szepesvári, C., and Willemson, J., "Improved Monte-Carlo Search," Tech. Rept., Univ. of Tartu, Vol. 1, Tartu, Estonia, 2006.
- [125] Kohlman, L. W., and Patterson, M. D., "System-Level Urban Air Mobility Transportation Modeling and Determination of Energy-Related Constraints," *2018 Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2018-3677, June 2018.
- [126] Patterson, M. D., Antcliff, K. R., and Kohlman, L. W., "A Proposed Approach to Studying Urban Air Mobility Missions Including an Initial Exploration of Mission Requirements," *75th Annual Forum and Technology Display*, May 2018.
- [127] Anon., "Google Maps," 2020, <https://www.google.com/maps> [retrieved 15 Feb. 2020].
- [128] Syed, N., Rye, M., Ade, M., Trani, A., Hinze, N., Swingle, H., Smith, J. C., Dollyhigh, S., and Marien, T., "Preliminary Considerations for ODM Air Traffic Management Based on Analysis of Commuter Passenger Demand and Travel Patterns for the Silicon Valley Region of California," *17th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA Paper 2017-3082, June 2017.  
<https://doi.org/10.2514/6.2017-3082>
- [129] Vascik, P. D., Cho, J., Bulusu, V., and Polishchuk, V., "A Geometric Approach Towards Airspace Assessment for Emerging Operations," *13th USA/Europe Air Traffic Management Research and Development Seminar*, June 2019.
- [130] Anon., "Near Midair Collision Reporting," 2017, <http://www.faraim.org/aim/aim-4-03-14-530.html> [retrieved 15 Feb. 2020].
- [131] Yang, X., and Wei, P., "Multi-aircraft Guidance and Separation Assurance," 2020, [https://github.com/xuxiyang1993/Multi\\_MCTS\\_Guidance\\_Separation\\_Assurance](https://github.com/xuxiyang1993/Multi_MCTS_Guidance_Separation_Assurance) [retrieved 15 Feb. 2020].