Autonomous Planning for Targeted Observation of Severe Weather

Michael Moncton[1], Himanshu Gupta[1], Zachary Sunberg[1], and Eric Frew[1]

University of Colorado Boulder, Boulder, CO 80309, USA
{michael.moncton,himanshu.gupta,zachary.sunberg,eric.frew}@colorado.edu

Abstract. This paper describes the decion-making architecture for an autonomous airborne scientist that closes planning loops over high fidelity weather models. The autonomy architecture exploits: local computing on the aircraft; edge computing in the field; and cloud computing accessible through the Internet, and other sensing and computing at fixed sites. This paper describes strategic-level information-space planning for determining where to fly and when to release air-launched drifters and tactical-level motion planning in complex flow fields.

Keywords: Targeted observation · UAS · Autonomous decision-making.

1 Introduction

Aerial robotic systems operating in complex environments will need to reason over high-fidelity models and large amounts of data. Aerial robots are fundamentally limited by size, power, and weight constraints. These constraints limit the payload and thus computational power that can be carried by any single robot. While there is a tendency to make these robots self-contained, that approach limits system capability. By leveraging computing dispersed across different robots, located in ground control stations in the field, or housed in high-performance computing centers, significantly more complex prediction, learning, and planning algorithms can be run compared to what could be implemented locally onboard.

Cloud robotics [11,19], the Internet of robotic things [15], and the dynamic data-driven application systems paradigm [5,1] are related concepts whereby networked communication bridges the gap between dispersed computational and physical elements. This paper describes an architecture built on networked command and control with dispersed computing that has been deployed [4] for a variety of applications that include coordinating uncrewed aircraft tracking moving targets [3], accessing numerical weather predictions within UAS planning algorithms from the field [7] and sampling severe thunderstorms from multiple unmanned aircraft simultaneously [2,9].

The motivating application for this paper is targeted observation of a supercell thunderstorm to understand tornado formation. The region of the storm hypothesized to influence tornado formation [13] is 32,000 m by 8,000m by 1,600m.

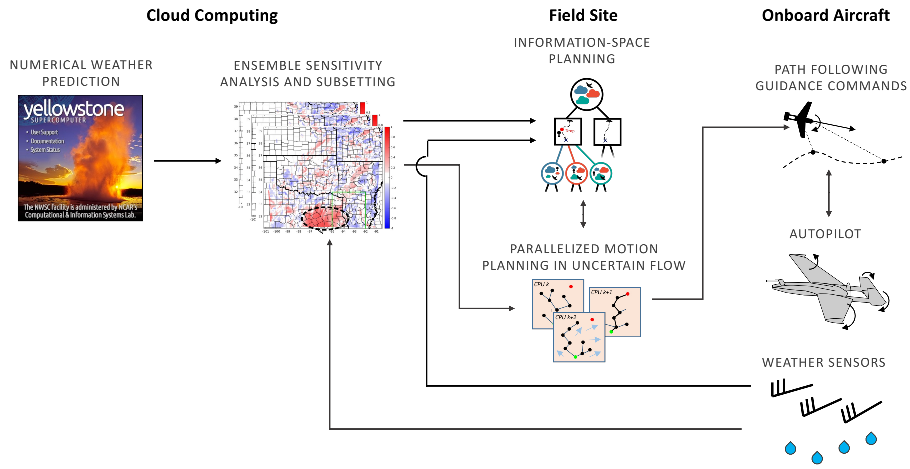


Fig. 1: The autonomy architecture exploits local computing on the aircraft, edge computing in the field on a mobile command center, cloud computing accessible through the Internet, and other sensing and computing at fixed sites.

Assuming SUAS speed of 20 m/s and sensing resolution of 1 m^2, it would take over 650 years to sample this region. Since this type of exhaustive sampling is not feasible, even with hundreds of aircraft, targeted observation schemes are needed in which online measurements guide the sensing system to good locations to improve model forecasts and to increase scientific understanding.

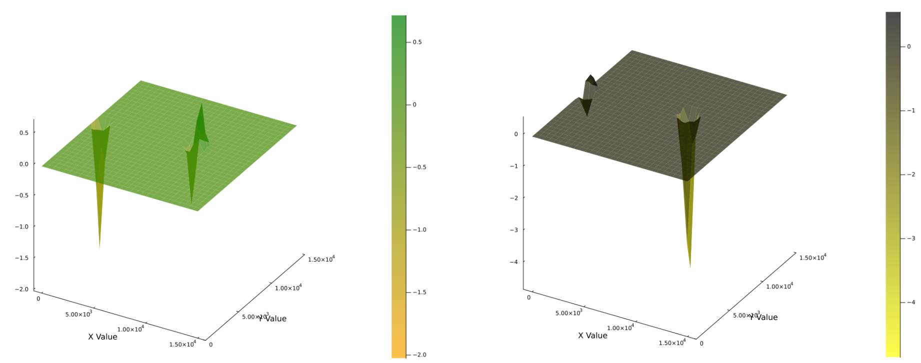
The focus of this paper is on the decision-making architecture for a new type of *autonomous airborne scientist*. This architecture fits the dynamic data driven application system paradigm by closing decision-making loops over ensembles of high fidelity weather models (Figure 1). Furthermore, this dispersed autonomy architecture exploits local computing on the aircraft, edge computing in the field on a mobile command center, cloud computing accessible through the Internet, and other sensing and computing at fixed sites. Because severe storm models cannot be run online fast enough for decision-making loops yet, ensemble subsetting is used to create a set of weather models for planning. These models are used at runtime for planning. As measurements are acquired, the decision-making loop dynamically updates its probabilistic understanding of the model subsets and uses this new understanding to control further path planning.

This paper describes steps toward online planning across dispersed computing in the context of targeted observation of complex atmospheric flows, by describing strategic-level information-space planning for determining where to fly and when to release air-launched drifters [18] and tactical-level motion planning in complex flow fields.

2 Strategic Level: Information Space Planning

Autonomous robots encounter uncertainty across all fields. This uncertainty in the robot's environment can be introduced by factors like noise in dynamics or measurements, and shifting unstructured surroundings. Such environments are characterized as being partially observable, where the robot cannot accurately perceive the true state of the environment. In the context of severe weather observation, the primary uncertainty is in the state of the weather system. The Partially Observable Markov Decision Process (POMDP) provides a principled mathematical framework for modeling aleatoric uncertainty in the outcomes of actions and epistemic uncertainty resulting from only partial observability of the state of the environment. Unfortunately finding the exact solution to POMDPs is computationally intractable [14], so we solve them approximately.

Recent efforts to solve POMDPs focus on using online sampling-based tree search techniques [17,20]. These tree searches are guided by maintaining value estimates over every node in the tree which are generally initialized by running a roll-out policy [8]. The overall system works as follows. The agent maintains a belief over the state space which measures the likelihood of a particular state being the true state of the environment at that time. Starting from the current belief, the agent builds a tree that helps it reason over the possible future outcomes and choose the best action within a fixed computation budget. The agent executes that action, receives a new observation, and updates its belief. This repeats until the termination criterion is met. Although building the belief tree to find the best action seems straightforward, its computational complexity increases exponentially with the number of states, actions, observations, and the planning horizon.



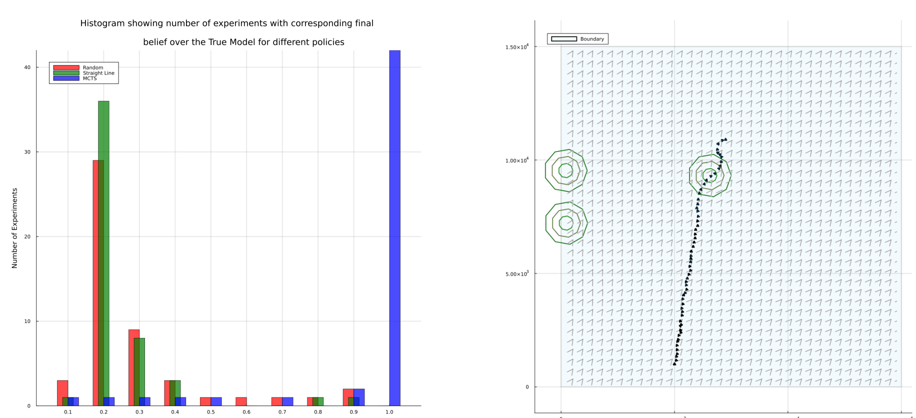
(a) Difference in temperature values of two different models.

(b) Difference in pressure values of two different models.

Fig. 2

For applications such as numerical weather prediction, the objective is to reduce uncertainty about the state of the environment, rather than maximize the expected reward. This class of problems is a generalization of the POMDP where the cost may be an arbitrary function, such as the entropy, of the belief [16]. In this work, we assume we have access to an ensemble of weather models and need to identify which model from the ensemble best explains the observed data. To achieve this, we need to identify regions in the environment where the SUAS should fly to gather observations that will minimize the entropy of our belief over the weather models.

For our experimental purposes, we generate seven different weather models that make up the ensemble. Each weather model generates only three quantities of interest: wind, temperature, and pressure. We sample three regions at random in a square field of size 10000m × 10000m where the model outputs are significantly different (Fig. 2). Flying through these regions is essential to identify the true weather model. We compare three policies for this problem: a random policy that takes random actions, a straight-line policy that flies the SUAS straight, and a tree-search policy (MCTS). Our preliminary results from fifty different experiments are summarized in Figure 3. We observe that the MCTS policy is able to identify the true weather model from the ensemble of seven models in more than 90% of the experiments. In contrast, both random and straight-line policies struggle a lot. Moreover, MCTS policy can identify which high-information regions are feasible to reach under the given wind pattern as shown in Figure 3.b. Our ongoing work focuses on using the Weather Research and Forecasting (WRF) model to obtain an ensemble of realistic weather models and deploying the SUAS in severe weather conditions to validate the effectiveness of our approach.



(a) POMDP planning performance compered to simple heuristics.

(b) High level path executed by the SUAS using POMDP planning.

Fig. 3: POMDP planning results

3 Tactical Level: Sample-based Motion Planning in Flow

Given the high-level strategic plan, tactical-level planning rapidly determines trajectories that are feasible given the nonlinear dynamics of the aircraft in complex flow fields. Access to dispersed computing opens the possibility for sample-based methods that can i.) forward-propagate nonlinear system dynamics; ii.) use importance sampling to capture uncertainties in the system [6], and iii.) be parallelized. A standard method for achieving asymptotic optimality in motion planning is through path rewiring [10]. For vehicles with complex, nonlinear dynamics models, rewiring is computationally prohibitive since it requires solving a two point boundary value problem. Instead, this work uses a planner that rapidly and efficiently explores the domain without rewiring.

The Local Optimality Grid (LOG) planner (Algorithm 1) is a lightweight planning algorithm that utilizes the grid provided by the wind field prediction or weather forecasting subsystem as a basis for storing information about nodes of the graph that are regionally optimal. It fundamentally consists of three components: selecting a node (Algorithm 2), propagating that node, and evaluating the result (Algorithm 3).

Speed improvements are made through quick and intelligent node selection for dynamics propagation. Such improvements are mainly gained from minimizing nearest neighbor searches (NNS) which become expensive as graphs expand. Intelligent node selection is facilitated through the use of a data structure similar to the structure of the wind field. This data structure stores information on nearby nodes with the best costs and is quick to build and access.

Algorithm 1 LOCAL OPTIMALITY GRID

for N *iterations* **do**
  $x_{prop} \leftarrow$ STATE_SELECTION(X)
  $x_{new} \leftarrow$ DYNAMICS_PROPAGATION(x_{prop})
  if COLLISION_FREE(x_{prop}, x_{new}) **then**
    EVALUATE_NODE(x_{new})

Algorithm 2 STATE_SELECTION(X)

$x_{rand} \leftarrow$ STATE_SAMPLE()
$c_{local} \leftarrow$ LOCAL_CELL(x_{rand})
if c_{local} is populated **then return** rep of c_{local}
for neighbor of c_{local} **do**
  if neighbor is populated **then return** rep of neighbor
return nearest active node

The LOG algorithm was compared against an SST algorithm [12], a similar forward-propagating planner that avoids rewire functions. Each planner was run in identical environments using the same dynamics propagation and control space. Graph data structures and NNS algorithms were also the same for both implementations as they are fundamental to planner speed.

Algorithm 3 EVALUATE_NODE(x_{new})

$c_{local} \leftarrow$ LOCAL_CELL(x_{rand})
if c_{local} is unpopulated **then**
  populate c_{local} with x_{new}
  add x_{new} to graph and activate
else if $x_{new}.cost < c_{local}.rep.cost$ **then**
  make $c_{local}.rep$ inactive
  add x_{new} to graph and activate
  make x_{new} rep of c_{local}

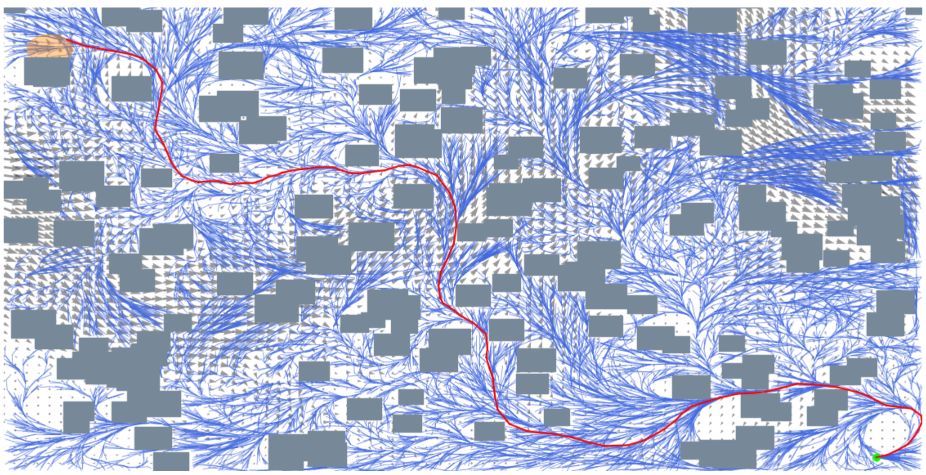
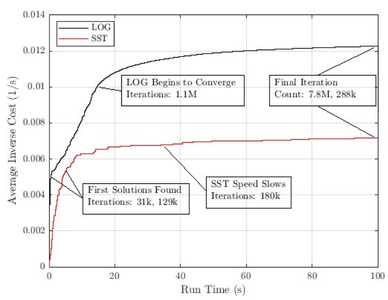


Fig. 4: LOG planner run including obstacles, wind field, graph, and solution path.

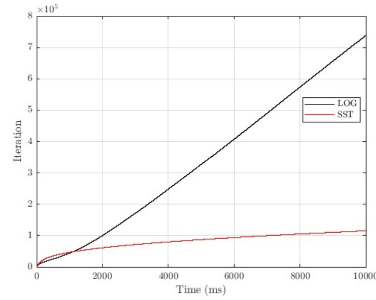
An environment of dense obstacles with a strong non-uniform wind field was used along with a kinematic unicycle model. Ninety instances of each planner were run for 100 seconds each controlling for initial RNG seed. Each algorithm was run single threaded on a 3.3 GHz AMD Ryzen 9 5900hx.

Planner results are shown in Fig. 5. Inverse solution cost averaged over all planner instances is tracked over time for both planning algorithms (left). Notable points in the algorithms progression are marked along with their respective iteration counts. The progression of planner iterations over time for both algorithms is shown (right). LOG finds solutions significantly faster than SST and continues to refine solution quality until approaching convergence. SST, however, is unable to find near-optimal trajectories in this environment as the large graph necessary to saturate the environment slows planner operation. The log planner is not slowed by large graph sizes, but rather speeds up as more of the enviornment is saturated and fewer NNS operations are required.

Although it appears that the two algorithms are converging upon different path costs, viewing the progression of both algorithms iteration-wise implies that SST has yet to converge, and practically is unable to in the given environ-



(a) Planner inverse solution cost over time averaged over all runs.



(b) Iterations over time for both planner instances

Fig. 5

ment. The LOG planner finds initial solutions in fewer iterations than SST and requires over one million iterations to begin to converge. SST, however, finds initial solutions slower and does not exceed 300 thousand iterations throughout the entire 100 second run. Therefore it is likely that SST is simply unable to converge in the given computation time and the asymptotic-appearing behavior of the average inverse cost is simply because the planner has slowed too much to make significant progress in optimizing solutions.

4 Conclusion

This paper described the decion-making architecture for an autonomous airborne scientist that closes planning loops over high fidelity weather models. The strategic-level planner uses a Partially Observable Markov Decision Process (POMDP) to formulate a problem which is solved using Monte Carlo Tree Search. The tactical planner uses a new Local Optimality Grid (LOG) planner to determine feasible motion plans. Both levels use sample-based methods which are amenable to parallelization as part of future work.

References

1. Bird, J., Glasheen, K., Frew, E.: Integrated planning, decision-making, and weather modeling for uas navigating complex weather. In: DDDAS2020: InfoSymbiotics/Dynamic Data Driven Applications Systems. Boston, MA (September 2020)
2. Frew, E., Argrow, B., Borenstein, S., Swenson, S., Hirst, A., Havenga, H., Houston, A.: Field observation of tornadic supercells by multiple autonomous fixed-wing drones. Journal of Field Robotics **37**(6), 1077–1093 (2020)
3. Frew, E., Dixon, C., Borenstein, S., Glasheen, K., Rajasekaran, R.K., Watza, S., Mills, A.B.: Lessons learned from field testing swarming unmanned aircraft at the university of colorado. In: AUVSI Xponential. Denver, CO (May 2018)

4. Frew, E., Glasheen, K., Hirst, A., Bird, J., Argrow, B.: A dispersed autonomy architecture for information-gathering drone swarms. In: IEEE Aerospace Conference. Big Sky, MT (March 2020)
5. Frew, E.W., Argrow, B., Houston, A., Weiss, C., Elston, J.: An energy-aware airborne dynamic data-driven application system for persistent sampling and surveillance. In: International Conference on Computational Science. Barcelona, Spain (June 2013)
6. Glasheen, K., Bird, J.J., Frew, E.W.: Experimental assessment of chance-constrained motion planning leveraging dispersed computing for small uncrewed aircraft. Field Robotics (2023)
7. Glasheen, K., Steiner, M., Pinto, J., Frew, E.: Experimental assessment of local weather forecasts for small unmanned aircraft flight. AIAA Journal of Aerospace Information Systems (2019)
8. Gupta, H., Hayes, B., Sunberg, Z.: Intention-aware navigation in crowds with extended-space pomdp planning. arXiv preprint arXiv:2206.10028 (2022)
9. Hirst, C.A., Bird, J., Burger, R., Havenga, H., Botha, G., Baumgardner, D., DeFelice, T., Axisa, D., Frew, E.: An autonomous uncrewed aircraft system performing targeted atmospheric observation for cloud seeding operations. Field Robotics **3**, 687–724 (2023)
10. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research **30**(7), 846–894 (2011). `https://doi.org/10.1177/0278364911406761`
11. Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A Survey of Research on Cloud Robotics and Automation. IEEE Transactions on Automation Science and Engineering (T-ASE) **12**(2) (2015)
12. Li, Y., Littlefield, Z., Bekris, K.E.: Asymptotically optimal sampling-based kinodynamic planning. International Journal of Robotics Research **35**(5), 528–564 (2016). `https://doi.org/10.1177/0278364915614386`
13. Orf, L., Wilhelmson, R., Lee, B., Finley, C., Houston, A.: Evolution of a long-track violent tornado within a simulated supercell. Bulletin of the American Meteorological Society **98**(1), 45–68 (2017). `https://doi.org/10.1175/BAMS-D-15-00073.1`
14. Papadimitriou, C.H., Tsitsiklis, J.N.: The complexity of Markov decision processes. Mathematics of Operations Research **12**(3), 441–450 (1987)
15. Ray, P.P.: Internet of Robotic Things: Concept, Technologies, and Challenges. IEEE Access **4**, 9489–9500 (2016). `https://doi.org/10.1109/ACCESS.2017.2647747`
16. Sunberg, Z., Chakravorty, S., Erwin, R.S.: Information space receding horizon control for multisensor tasking problems. IEEE Transactions on Cybernetics **46**(6), 1325–1336 (2016), `http://ieeexplore.ieee.org/document/7174988/`
17. Sunberg, Z., Kochenderfer, M.J.: Online algorithms for POMDPs with continuous state, action, and observation spaces. In: International Conference on Automated Planning and Scheduling (2018)
18. Swenson, S., Argrow, B., Frew, E., Borenstein, S., Keeler., J.: Development and deployment of air-launched drifters from small uas. Sensors **19**(2149), 1–16 (2019)
19. Wan, J., Tang, S., Yan, H., Li, D., Wang, S., Vasilakos, A.V.: Cloud robotics: Current status and open issues. IEEE Access **4**, 2797–2807 (2016). `https://doi.org/10.1109/ACCESS.2016.2574979`
20. Ye, N., Somani, A., Hsu, D., Lee, W.S.: DESPOT: Online POMDP planning with regularization. Journal of Artificial Intelligence Research **58**, 231–266 (2017)