A Mixed-Integer Programming Formulation for Multi-Agent Motion Planning of an Optimally Sized Line of Sight Network

Collin Hudson
Aerospace Engineering Sciences
University of Colorado Boulder
collin.hudson@colorado.edu

***Abstract*—This work outlines the development of a multi-agent path planning system with the goal of maintaining line of sight between a set of Uncrewed Aircraft Systems (UAS) and ground agents to form a continuous communication network in a cluttered environment. In order to reduce the number of UAS required for missions, which increase costs and operator requirements, the problem is presented as an optimization problem to find the smallest connected network at each time step that maintains connectivity between all ground agents. The problem is formulated as a Mixed-Integer Program with constraints that ensure all agents respect obstacles, maintain a connected network, and produce viable paths with the objective of minimizing the number of UAS agents in the network. Solutions to the problem were generated using the SCIP optimizer and demonstrated via simulation using the Julia programming language.**

I. Introduction

Communication often relies on signal line of sight (LOS) between the transmitter and receiver, which can be obstructed by terrain or other obstacles and affect coordination efforts between teams, such as multiple search parties operating in a mountainous region that need to be able to coordinate search patterns. This limitation can be alleviated by forming a network of small UAS that transmit signals through the network between parties. In order for the UAS network to enable communication, it must maintain a connection between all parties through LOS with either a UAS or other party for the entire mission. Since communication could be between teams or other ground stations, any transmitters or receivers that need to maintain a connection will be referred to as ground agents (GA). Essentially, for the entire mission every ground agent must be connected to every other ground agent by either having LOS with at least one other ground agent or UAS that is in LOS with the rest of the UAS network. The number of UAS agents used to establish this network directly relates to the operational cost of the mission, due to both the direct cost of and the required personnel to launch and, in the case of non-autonomous aircraft, operate each UAS. Therefore, it is highly beneficial to minimize the number of UAS agents necessary for a given mission. Given that any optimization must reason about which connections to maintain among a discrete number of agents, the optimization problem can be suitably formulated as a Mixed-Integer Program, a method that is well studied for motion planning of Multi-Agent Systems [1], particularly with respect to fully connected multi-agent networks [5]. The formulation presented here builds upon work done to formulate LOS constraints with sample points [2] to ensure valid connections are being made between agents. In addition, a formulation for the Minimum Connectivity Inference (MCI) problem [3], in which the number of edges is minimized while guaranteeing that a subset of vertices are connected, is incorporated to ensure the generated network connects all ground agents together. While there are formulations that utilize Constraint Generation to solve the MCI problem faster than MILP formulations [4], the large number of constraints on connections between agents makes the repeated solving inherent to the approach much less desirable than the MILP approach.

II. Problem Formulation

The problem can be summarized as follows: "Given the positions of n_{ga} ground agents at each time step, find trajectories for at most n_{uas} available UAS agents that connect all ground agents for all time steps in a line of sight network". These trajectories must respect all obstacles in the environment, which are modeled as convex polygons. Let the total number of obstacles in the environment equal n_{obs}, and the sum of the number of edges of all obstacles in the environment equal n_e. The problem is presented as a Mixed-Integer Nonlinear Programming problem at every time step for a finite number of time steps. The state is represented as follows:

- x_i, x coordinate of agent i.
- y_i, y coordinate of agent i.
- $c_{i,j}$, binary variable for connection between agents i and j.
- $o_{h,i}$, binary variable for obstacle half-space constraint h for agent i.
- $l_{h,i,j,k}$, binary variable for obstacle half-space constraint h for sample point k on line between agents i and j.
- a_i, binary variable for if agent i is in the network.
- $f_{i,j}$, flow from agent i to agent j.

We assume that the position $\mathbf{p}$ of each ground agent is known at each time step, leading to the following constraint.
Let the position of ground agent i at time step k be specified by $g(i,k)$. Let the index sets for ground agents, UAS agents,

obstacles, and obstacle edges be represented by $\mathcal{G}$, $\mathcal{U}$, $\mathcal{O}$, and $\mathcal{O}_e$, respectively.

$$\mathbf{p}_{i,k} = g(i,k) \quad \forall i \in \mathcal{G} \tag{1}$$

A. Connection Constraints

The connection of two agents is limited by some communication radius r_{con} as an analog to real-world limits on signal strength. This constraint is implemented using the Big M method such that the constraint is completely relaxed if two agents are not connected.

Let $M_c = 1.1((x_{max} - x_{min})^2 + (y_{max} - y_{min})^2)$, where the minimum and maximum values are the bounds of the environment. The constraint is defined as follows:

$$||\mathbf{p}_i - \mathbf{p}_j||^2 \leq r_{con}^2 + M_c(1 - c_{i,j}) \quad \forall i,j \in \mathcal{G} \cup \mathcal{U} \tag{2}$$

Thus, if agent i and j are not connected the inequality becomes trivial, since x and y are constrained to be within the bounds of the environment so their difference can never exceed M_c. As a result of this constraint, the Mixed-Integer Program is quadratic.

B. Obstacle Avoidance Constraints

We used the MIP formulation for obstacle avoidance mentioned in [1] to find optimal Big M values for each obstacle half-space, and ensure that no agents collide with the obstacles in the environment. The following linear program is solved for Big M value $M_i = (k_i - \mathbf{h_i}^T\mathbf{p})$ for half-space i, where $\mathbf{h_i}$ and k_i define the plane $\mathbf{h_i}^T\mathbf{p} = k_i$.

$$\begin{aligned}
&\underset{\mathbf{p}}{\text{Maximize}}\ (k_i - \mathbf{h_i}^T\mathbf{p}) \\
&\text{Subject to:} \\
&\mathbf{p} \in \text{environment bounds}
\end{aligned}$$

Then, the following constraints are added to our formulation. Let $\{E_j\}$ equal the set of indices of half-spaces corresponding to obstacle j. In order to reduce the number of constraints, it is assumed that all ground agents do not collide with any obstacles for all time steps.

$$-\mathbf{h_j}^T\mathbf{p}_i \leq -k_j + M_j(o_{j,i}) \quad \forall i \in \mathcal{U}, j \in \mathcal{O}_e \tag{3}$$

$$\sum_{k \in E_j} o_{k,i} \leq |E_j| - 1 \quad \forall i \in \mathcal{U}, j \in \mathcal{O} \tag{4}$$

Constraint 4 ensures that at least one constraint 3 for obstacle j is enforced, such that agent i is within at least one external half-space of obstacle j.

C. Network Constraints

In the formulation for the Minimum Connectivity Inference (MCI) problem discussed in [3], the subset of nodes, called a cluster, to be connected is known. In our problem, the number of UAS agents is part of the state and must be optimized and is therefore unknown with respect to the MCI problem. The following constraints are derived from the formulation introduced in [3], with some modifications to fit our problem. We introduce the binary variable a_i to encode whether agent i is in the cluster we are evaluating for minimal connectivity. A matrix of non-zero values, f, is added to the state to represent flow between agents. Summing over column i will give the total flow into agent i. Similarly, summing over row i gives the total flow out of agent i. For simplicity, we assume that ground agent 1 is the source node. The following constraints are added.

$$\sum_j \sum_i c_{i,j} \geq \sum_i (a_i) - 1, \tag{5}$$

$$\sum_j c_{i,j} \leq (n_{ga} + n_{uas})a_i \quad \forall i \in \mathcal{G} \cup \mathcal{U}, \tag{6}$$

$$\sum_i c_{i,j} \leq (n_{ga} + n_{uas})a_j \quad \forall j \in \mathcal{G} \cup \mathcal{U}, \tag{7}$$

$$\sum_j f_{i,j} \leq (n_{ga} + n_{uas})a_i \quad \forall i \in \mathcal{G} \cup \mathcal{U}, \tag{8}$$

$$\sum_j f_{i,j} - \sum_k f_{k,i} = -a_i \quad \forall i \neq 1 \in \mathcal{U}, \tag{9}$$

$$f_{i,j} - f_{j,i} \leq (\sum_i (a_i) - 1)c_{i,j} \quad \forall i < j \in \mathcal{G} \cup \mathcal{U}, \tag{10}$$

$$f_{i,j} \geq 0 \quad \forall i = j \in \mathcal{G} \cup \mathcal{U} \tag{11}$$

Constraint 5 ensures that there are enough connections to connect all agents in the in the network. Constraints 6 and 7 ensure that agents must be in the network to have any connections. Constraint 8 ensures that agents must be in the network to have any flow out of them. Constraint 9 specifies that one unit of flow must be consumed when passing through an agent in the network, unless it is the source node (ground agent 1). Constraint 10 requires no net flow between unconnected agents, and that connected agents have a net flow less than the number of agents in the network. Constraint 11 defines flow as between different agents. In addition, since the ground agents must be connected in order to have a valid network, the following constraint is added.

$$a_i = 1 \quad \forall i \in \mathcal{G} \tag{12}$$

D. LOS Constraints

In order to ensure that connections only occurred between agents within LOS of each other, we utilized the MIP formulation introduced in [2]. The key concept behind the formulation is the introduction of a sample point (or points) between two nodes, with the nodes being considered in LOS if the sample point belongs to both of the external half-spaces that contain each node. For our formulation, we used the midpoint $\mathbf{m_{i,p}}$ of the line between agents i and k as the sample point. The following constraints are added:

$$-\mathbf{h_j}^T\mathbf{m_{i,p}} \leq -k_j + M_j(o_{j,i}) + M_j(l_{j,i,p,1}) \quad \forall i,p \in \mathcal{G} \cup \mathcal{U}, j \in \mathcal{O}_e \tag{13}$$

$$-\mathbf{h_j}^T\mathbf{m_{i,p}} \leq -k_j + M_j(o_{j,p}) + M_j(l_{j,i,p,1}) \quad \forall i,p \in \mathcal{G} \cup \mathcal{U}, j \in \mathcal{O}_e \tag{14}$$

$$c_{i,p} \leq 1 - l_{j,i,p,1} \quad \forall i,p \in \mathcal{G} \cup \mathcal{U}, j \in \mathcal{O} \tag{15}$$

Constraint 15 ensures that if agents i and p are connected, the LOS constraint must not be relaxed. While the original formulation uses multiple sample points, our formulation was only reliable when using the midpoint as the sample point, leading to a very conservative constraint on LOS between agents.

E. Time Step Constraints

In order for the solution paths to be viable, UAS agents must be able to reach their specified position at the next time step. A radius constraint is added for each UAS agent that was active in the last time step, with the radius corresponding to an assumed maximum UAS velocity. This is implemented using the Big M value derived for the connectivity constraints for each time step $k > 1$.

$$||\mathbf{p}_{i,k} - \mathbf{p}_{i,(k-1)}||^2 \leq r_{uas}^2 + M_c(1 - a_{i,(k-1)}) \quad \forall i \in \mathcal{U} \quad (16)$$

F. Objective Function

The overall objective is to reduce the number of agents in the network while maintaining ground agent connectivity. In addition, the network should use the UAS agents that were in the network at the previous network if possible in order to maintain path viability. Finally, in order to reduce the number of unnecessary connections, the network should use the least amount of edges between UAS agents as possible. The problem at time step k is finally formulated as follows.

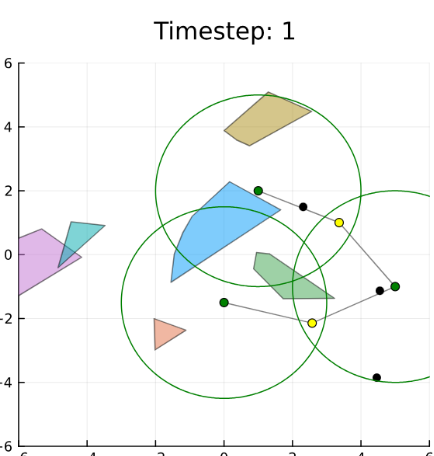
$$\underset{\mathbf{x}}{\text{Minimize}} \sum_i a_{i,k} + \sum_i (a_{i,k} - a_{i,(k-1)})^2 + 0.1 \sum_{j \in \mathcal{U}} \sum_{i \in \mathcal{U}} c_{i,j,k}$$

Subject to:

Constraints 1-16

III. Simulation

To solve the optimization problem at each time step, the problem was formulated using the JuMP package for the Julia programming language using the off-the-shelf optimizer SCIP [6]. The SCIP optimizer was chosen for its ability to handle Mixed-Integer Problems with quadratic constraints. An example trajectory plan is shown below, with ground agents shown in green, active UAS agents in yellow, and inactive UAS agents in black.



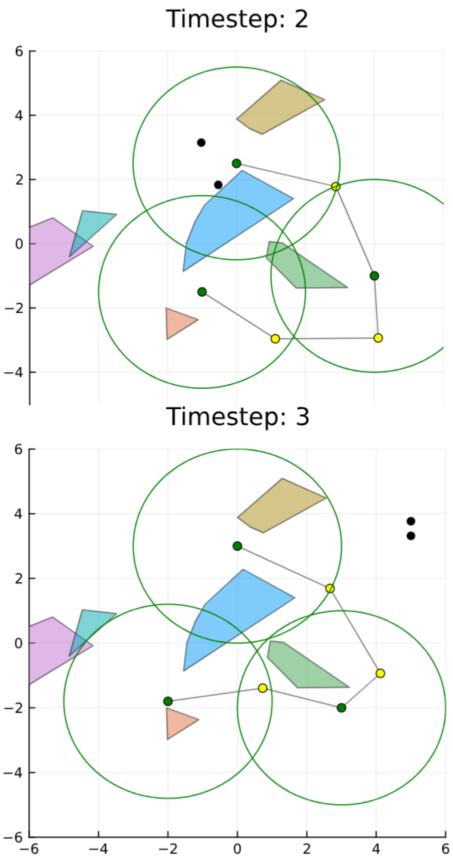


Fig. 1: Trajectory plan for minimal network

As shown in Figure 1, the planner is able to find connected networks at each time step that respect all obstacles. In addition, the planner can add UAS agents to the network when necessary, as shown in time step 2. Monte Carlo simulations were also performed to analyze the effect of the number of obstacles, ground agents, or available UAS agents on the performance of the system for a single time step. To analyze the effect of obstacles on the speed and network size, random environments of n obstacles and the positions of the ground agents were generated for three ground agents and five available UAS agents.

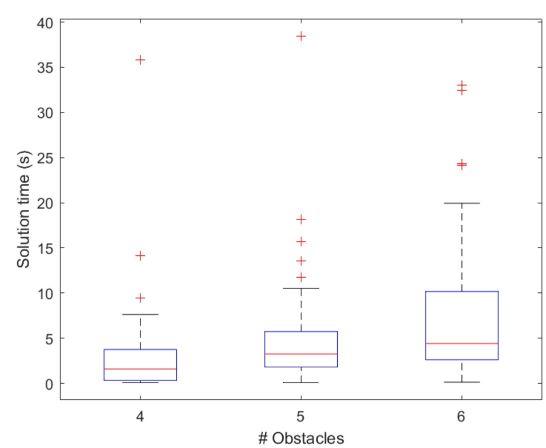


Fig. 2: Impact of obstacles on time to solution

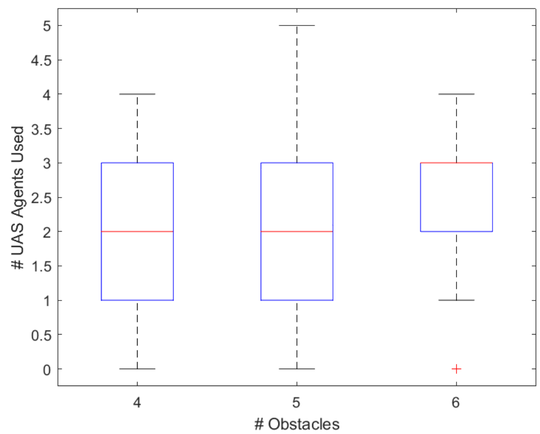


Fig. 3: Impact of obstacles on network size

As expected, the time to a solution increases with the number of obstacles. However, the size of the network is slightly more robust to increased obstacles. The network sizes are grouped around two to three UAS agents as the number of obstacles increases. For simulations with varied ground agents, random environments with five obstacles and available UAS agents were generated.

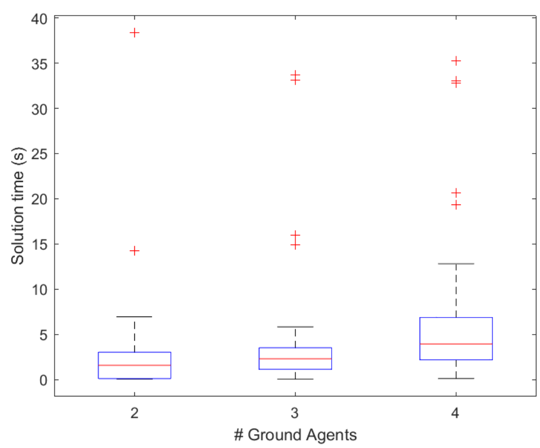


Fig. 4: Impact of ground agents on time to solution

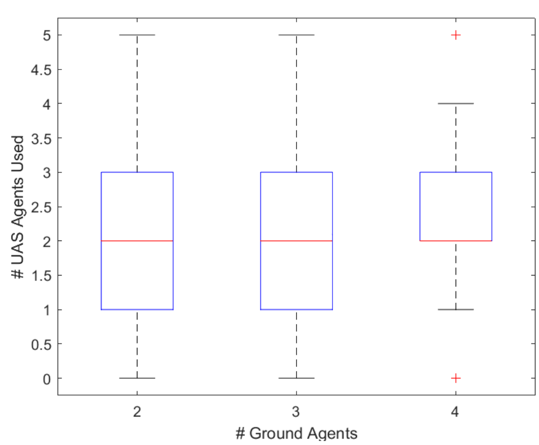


Fig. 5: Impact of ground agents on network size

The simulation time and network size follow a similar trend as the obstacle simulations. An interesting result is that the network size decreases slightly when the fourth ground agent is added. This can be explained by the fact that the environment is bounded, so there is a greater chance that ground agents can connect to each other directly as more ground agents are added. Lastly, the effects of increasing the number of available UAS agents was analyzed. Unsurprisingly, the simulation time increases as more available agents are added. However, it should be noted that the addition of UAS agents impacts the time to solution less than the addition of ground agents, which is an important result as networks are more likely to be comprised of many UAS agents for a smaller number of ground agents.

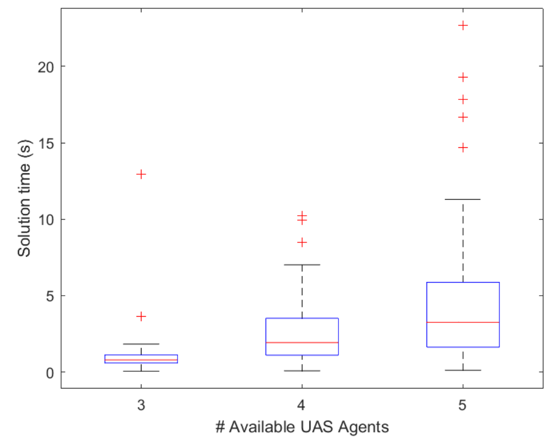


Fig. 6: Impact of UAS agents on time to solution

The rate at which the system found a valid solution while running the Monte Carlo simulations for each parameter is given below, with the cardinality of the simulation corresponding to the column shown on the respective bar plots shown above.

TABLE I: Monte Carlo Success Rate (%)

	Simulation		
Parameter	**1st**	**2nd**	**3rd**
Obstacle	100	92	92
Ground Agent	100	98	86
UAS Agent	86	98	98

Some important features to note are the increase in success rate with increasing available UAS agents, and decrease in success rate with an increasing number of ground agents. Due to the limited connection radius, if two ground agents are far enough apart, there must be enough UAS agents to span the distance between them. By increasing the number of available UAS agents, the optimizer is able to handle more dispersed ground agent configurations.

IV. Conclusions

The multi-agent LOS network problem for ensuring connectivity between a set of agents in a cluttered environment was successfully solved using a Mixed-Integer Program approach and the SCIP optimizer for constraint integer programming. While valid solutions were demonstrated, further work could investigate the actual optimality of the generated paths with respect to the minimum possible network size. In addition, further constraints could be placed on the UAS agent motion, ensuring that agents do not cross obstacles or collide with each other during transitions between time steps. Higher fidelity models for UAV motion could also be incorporated such that generated paths respect the kinematics and dynamics of each agent. Lastly, exact constraints for line of sight could be investigated, in order to better account for obstacles without limiting otherwise valid solutions based on the more conservative assumptions made in this formulation.

V. Contributions and Release

The author implemented the above formulation from scratch using the Julia programming language and JuMP.jl package, which was solved using the off-the-shelf optimizer SCIP. The author also implemented a random environment generator from scratch using the LazySets.jl package for random polygon generation. **The authors grant permission for this report to be posted publicly.**

References

[1] Daniel Ioan, Ionela Prodan, Sorin Olaru, Florin Stoican, Silviu-Iulian Niculescu. “Mixed-integer programming in motion planning”. Annual Reviews in Control, 2020, 10.1016/j.arcontrol.2020.10.008.hal-03108529,https://centralesupelec.hal.science/hal-03108529v1.

[2] A. Caregnato-Neto, M. R. O. A. Maximo and R. J. M. Afonso, “A Novel line of sight Constraint for Mixed-Integer Programming Models with Applications to Multi-Agent Motion Planning,” 2023 European Control Conference (ECC), Bucharest, Romania, 2023, pp. 1-6,https://doi.org/10.23919/ECC57647.2023.10178275.

[3] Dar, Muhammad Abid, Andreas Fischer, John Martinovic, and Guntram Scheithauer. 2018. “An Improved Flow-Based Formulation and Reduction Principles for the Minimum Connectivity Inference Problem.” Optimization 68 (10): 1963–83. https://doi.org/10.1080/02331934.2018.1465944.

[4] Bonnet, É., Fălămaş, DE., Watrigant, R. (2019). “Constraint Generation Algorithm for the Minimum Connectivity Inference Problem”. In: Kotsireas, I., Pardalos, P., Parsopoulos, K., Souravlias, D., Tsokas, A. (eds) Analysis of Experimental Algorithms. SEA 2019. Lecture Notes in Computer Science(), vol 11544. Springer, Cham. https://doi.org/10.1007/978-3-030-34029-$2_1$2.

[5] Afonso RJM, Maximo MROA, Galvão RKH. “Task allocation and trajectory planning for multiple agents in the presence of obstacle and connectivity constraints with mixed-integer linear programming”. Int J Robust Nonlinear Control. 2020; 30: 5464–5491. https://doi.org/10.1002/rnc.5092

[6] Suresh Bolusani, Mathieu Besançon, Ksenia Bestuzheva, Antonia Chmiela, João Dionísio, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Mohammed Ghannam, Ambros Gleixner, Christoph Graczyk, Katrin Halbig, Ivo Hedtke, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Dominik Kamp, Thorsten Koch, Kevin Kofler, Jurgen Lentz, Julian Manns, Gioni Mexi, Erik Mühmer, Marc E. Pfetsch, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Mark Turner, Stefan Vigerske, Dieter Weninger, Lixing Xu, The SCIP Optimization Suite 9.0, Available at Optimization Online and as ZIB-Report 24-02-29, February 2024