

# Optimized Transit Planning and Landing of Aerial Robotic Swarms

Thomas F. Dono and Timothy H. Chung

**Abstract**—This research explores the efficient and safe landing and recovery of a swarm of unmanned aerial vehicles (UAVs). The presented work involves the use of an overarching (centralized) airspace optimization model, formulated analytically as a network-based model with side constraints describing a time-expanded network model of the terminal airspace in which the UAVs navigate to one or more (possibly moving) landing zones. This model generates optimal paths in a centralized manner such that the UAVs are properly sequenced into the landing areas. The network-based model is “grown” using agent-based simulation with simple flocking rules. Relevant measures of performance include, e.g., the total time necessary to land the swarm. Extensive simulation studies and sensitivity analyses are conducted to demonstrate the relative effectiveness of the proposed approaches.

## I. INTRODUCTION

The rapid development and deployment of unmanned systems has led to their increased use in many mission areas and has expanded the future concepts and research arenas of relevance. In particular, the opportunities for low cost robotic systems and increased distributed and networked capabilities offer potential for operations involving large teams of such agents.

Recently, unmanned aerial vehicles (UAVs) have been at the forefront of extensive use in operational settings. In the current operational environment, at least one human operator is required to control a single UAV during flight. As this technology evolves, there will eventually be tens of UAVs controlled by a single human operator. The level of autonomy must increase appropriately in order to reduce operator workload and still allow for mission effectiveness and completion.

The employment of such large UAV teams encompasses a wide range of technological and operational challenges relevant to multiple mission areas, one of which is the optimized planning for transit and sequencing for landing of these large teams of UAVs. Consider the ongoing development of the Navy’s X-47B UAV, which has recently demonstrated its ability to successfully land autonomously in test environments as a precursor to carrier deck landings [1]. However, the algorithmic foundations for addressing this landing problem for *many* aerial agents in an efficient and effective manner remains a challenging one, largely due to the computational intractabilities of such optimization problems.

Thus, the main contributions of this work include the construction of a network-based mathematical program optimizing the transit and landing problem that can provide optimal solutions (or measures of the optimality gap). However, the

computational challenges limit the size of the airspace, and thus, the operational relevance, of the problems currently solvable using state-of-the-art optimization solvers. Hence, this paper also proposes a number of heuristic approaches which extend the solution space to more realistic airspace scales, so that the optimization model can provide meaningful transit paths for actual operational scenarios. Further, an additional contribution is the integration and comparison of these approaches with traditional agent-based methods, e.g., flocking of swarms, where the presented optimization models provide the optimal solutions as baselines for performance and the agent-based models leverage computational simplicity.

The remainder of this paper further develops this work by first reviewing the relevant literature in Section II. Section III formulates the multi-UAV routing and landing problem as a network-based optimization model, including definition of the cost function as the total time taken to land all aerial agents at their destinations. The computational complexity of this problem motivates the approaches presented in Section IV to permit tractability of solutions, followed by Section V, which investigates several illustrative scenarios with numerical investigations. The paper concludes with summary remarks and avenues for future research in Section VI.

## II. RELATED WORKS

The presented work is closely aligned with problems of airspace management and aircraft sequencing in the terminal environment, such as may be seen in air traffic control contexts. However, the significant differences are twofold. Firstly, these airspace or runway management approaches address the transit of aircraft (coarse spatial and temporal scales) separately from the scheduling of landings (fine spatial and temporal scales); however, optimized sequencing of the many aircraft early in the transit phase can surely impact the performance at the terminal phase, the integration of which is addressed by this paper and captured by the objective function of the total time to transit and land all aircraft. Secondly, many of the previous works are descriptive, rather than prescriptive, stochastic models that apply to single airport and runway operations, whereas the presented approach provides a rigorous formulation and proposed procedures for addressing multiple landing sites with large numbers of aircraft.

Peterson et al. [2] looked at descriptive models for describing the nature of the air traffic around Dallas Fort Worth International Airport. They use discrete time Markov and semi-Markov chains (with discrete 15 minute intervals) to determine the expected delays, and also include six different state spaces, and an arrival process that is either time-varying Poisson or deterministic to capture heavy and light traffic periods. Bolender and Slater [3] similarly used  $M/D/2$  and  $M/M/2$  queues

Thomas F. Dono is with the Marine Corps Warfighting Laboratory in Quantico, VA, USA thomas.dono@usmc.mil. Timothy H. Chung is an Assistant Professor of Systems Engineering at the Naval Postgraduate School, Monterey, CA, USA thchung@nps.edu.

to evaluate the performance of multiple runway operations, concluding that most airports are not running at maximum capacity and that the presented models are useful in predicting arrival performance. Using graph-theoretic approaches, Saraf and Slater [4] developed an Eulerian circuit-based model used for real time optimization of arrival scheduling to determine the best sequence of arriving aircraft. Alternatively, Boesel [5] used an object-oriented Monte Carlo simulation model that uses individual objects for the aircraft, where each aircraft object must travel along pre-designated tracks to aid in the computations.

More aligned with the presented work, Brooker [6] suggests the scheduling of aircraft arrivals hundreds of miles away, even prior to their take-off, thereby enabling the aircraft to arrive on time, with no holding and only small flight modifications. This work differs from the traditional priority queuing systems and  $M/G/1$  queues that are often used to predict performance in airport arrival systems. This approach can be translated into the landing of large groups of UAVs, adjusting their flight paths appropriately to enter directly into final approach with no holding or delay. More specifically, any delay is absorbed during transit and set up to approach.

The use of network models is not new to the study of this problem, where graph representations and specialized algorithms can be applied to solve these types of problems. Dell’Olmo [7] explores network models for the transit problem and try to include what they refer to as “free flight capacity;” however, they can only account for existing pre-defined airways rather than allow use of the entire volume of the airspace. The  $K$ -King model is presented by Artiouchine et al. [8], using a chess board in which a king can move in any direction on the board but only one space at a time. The king is then limited by rules, e.g., a deconfliction constraint, in which it is not able to get too close to a fellow king. The kings then need to go from their current positions to a designated corner of the chessboard, which can be thought of as a landing site and thus relevant to the presented work.

Agent-based models (ABMs) have also been used for many years and in many fields, including seminal work and extensions of Reynolds’ Boids [9]. For example, Conway [10] developed an agent-based model to mimic the air traffic control systems utilized by many major airports to observe the behavior of the system. She concluded that this was a good tool for analyzing the behaviors of air traffic in the terminal environment. More recently, rigorous algorithms describing flocking of agents have been prolific. Olfati-Saber [11] describes several algorithms that have been used for the flocking of agents in both free and restricted space, where free space reflects the absence of obstacles in the environment. Others, such as Yu et al. [12], expand on the flocking and obstacle avoidance behaviors with the use of “fuzzy logic” with good results, having the agents form up faster with better speed and distance control than previous algorithms.

The work presented in this paper leverages the advantages of both network-based optimization methods and agent-based methods described above, so as to improve the operational

realism and relevance of the proposed research.

### III. NETWORK-BASED OPTIMIZATION MODEL FORMULATION

The advantages of rigorous combinatorial optimization models include quantifiable measures of the optimality of solutions; such formulations also allow careful construction of the objective function and articulation of associated constraints. This section presents the mixed integer mathematical program that provides a representative optimization model of the routing and landing problem.

An illustrative operational scenario is described as follows. Consider a large team of unmanned aerial vehicles poised to return to various landing sites in an efficient and sequenced manner. Figure 1 graphically shows an example of a reference scenario with the collective UAV swarm preparing for recovery at three distant but relatively clustered landing zones. This scenario and additional reference scenarios described later are investigated in greater detail in Section V.

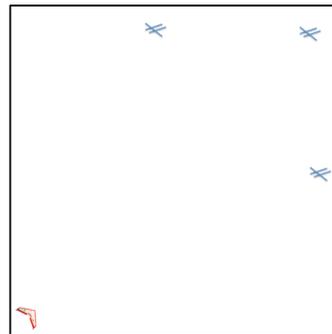


Fig. 1. Reference Scenario 1: UAV swarm (bottom left, red UAV icon) transits to three separate landing sites (upper and center right, blue airport markers)

We propose a network model with side constraints to describe the transit and landing of UAVs to landing sites, illustrated schematically in Figure 2. For illustration, we consider the problem in the 2D plane, where the continuous planar airspace is discretized and represented as a lattice graph. Note, however, that this model can easily be generalized to three dimensions, e.g., altitude in addition to latitude and longitude, which although increases the complexity of the problem, does not alter the underlying combinatorial optimization formulation.

Given the dynamic evolution of the transit and landing problem, the underlying graph with set of nodes,  $N$ , and corresponding data (such as capacity or availability of edges) may depend upon time. To address this temporal aspect, we construct a *time-expanded network model* [13], in which the graph is replicated for each discrete time step from 0 to a specified time horizon length,  $T$ , such that each node in the resulting network represents a spatial tuple along with a time index. For succinctness, the proposed network model with side constraints detailed in this paper is referred to as the network model or network-based model synonymously.

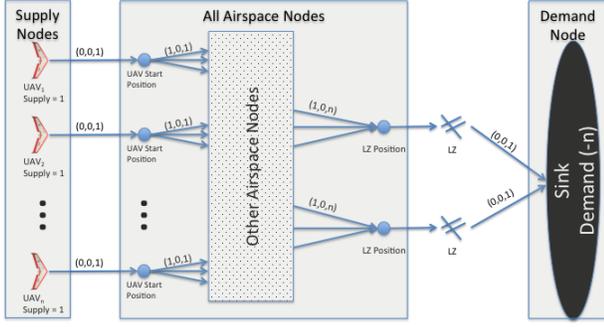


Fig. 2. Schematic of the network model, identifying supply, demand, and airspace nodes, and associated edges.

Consider each of the UAVs to initially start in nodes designated as “source node” or “supply node” (in the standard terminology for network models), such that at time step  $t = 0$ , these nodes supply the airspace with all UAVs in the swarm, with edges in the time-expanded network existing only for this initial time step. The initial locations of the UAVs within the airspace is denoted  $N_0 \subset N$ , where  $N$  represents discretized grid locations in Euclidian space.

Deconfliction of the airspace is addressed by requiring that a given node can contain at most a single UAV at any given time step, which relies on a correspondingly appropriate spatial decomposition of the airspace. The UAV is capable of reaching any adjacent nodes in a single time step, and cannot remain in the same node for consecutive time steps (such as for fixed wing aircraft).

Movement along an edge from node  $i \in N$  to node  $j \in N$  at time step  $t$ , denoted  $arc_{i,j,t}$ , is assumed to incur unit cost and expends one time step to complete, and in this manner, each UAV in the swarm moves through the network toward its assigned landing site  $a \in A$  out of multiple possible sites. These landing sites are additional nodes (replicated in time) that are reachable from the airspace via time-expanded edges in the network, which enable UAVs to land at a given landing zone at any time step throughout the mission. Once a UAV reaches a node corresponding to a landing site, it transits to a terminal “sink node,” which represents the “demand” for the entire network flow. Figure 2 graphically depicts this network-based formulation.

Given the above definitions, the network-based optimization can be formulated as the following mathematical program. Decision variable  $X_{i,j,t}$  represents flow of UAVs along  $arc_{i,j,t}$  between nodes  $i \in N$  and  $j \in N$  at time step  $t \in T$ ,  $C_{i,j}$  represents the cost of traveling from node  $i$  to node  $j$ ,  $b_{n,t}$  represents the flow balance constraint being zero for all nodes, with the exception of source and sink nodes,  $req_a$  is the minimum number of UAVs required to land at airport  $a$ ,  $\tau^a$  represents the inter-arrival separation required by airport  $a$ , and  $v_{i,j,t}$  is the upper bound capacity of  $arc_{i,j,t}$ .

$$\min_X \sum_{arc_{i,j,t}} C_{i,j} X_{i,j,t} \quad (1)$$

$$\text{s.t.} \sum_j X_{n,j,t} - \sum_i X_{i,n,t-1} = b_{n,t}, \forall n \in N, \forall t \in T \quad (2)$$

$$\sum_{t=t'}^{t'+\tau^a} X_{a,sink,t} \leq 1, \forall t' \in T, \forall a \in A \quad (3)$$

$$\sum_t X_{a,sink,t} \geq req_a, \forall a \in A \quad (4)$$

$$0 \leq X_{i,j,t} \leq v_{i,j,t}, \text{ integer}, \forall arc_{i,j,t} \quad (5)$$

The objective function (Equation 1) expresses the total cost of all UAVs traveling through the airspace. Recall that for uniform unit cost along each edge in the network, this objective represents the total time until all UAVs have landed at their designated landing sites. Each constraint described in Equation 2 is a flow balance equation, ensuring any supply that enters a given node must exit the node with the exception of the UAV and sink nodes. Constraints 3 limit the number of UAVs that can land in a given landing zone  $a$  within  $\tau^a$  time steps. Constraints 4 ensure that a minimum required number of UAVs,  $req_a$ , land at a given landing zone, and Constraints 5 ensure that  $v_{i,j,t}$ , is representing the inter-arrival separation required by airport  $a$  representing the inter-arrival separation required by airport  $a$  not exceeded.

Standard construction of the time-expanded network includes all locations in the planar airspace at all time steps, which results in a large, memory-intensive model. Given the constraint on computation, the above network model can be solved using standard solvers, e.g., GAMS/CPLEX [14], but only for limited airspace sizes. We quickly discover that there is an urgent need to reduce the state space by removal of any excess nodes and edges to make the problem more tractable. Without steps for node reduction, current computation limits prevent solving for airspaces in excess of  $60 \times 60$  nodes<sup>1</sup>. This, for our purposes, is operationally irrelevant as the airspace it represents is insignificantly small. We choose to use the number of nodes to assess computational complexity since there is a direct relationship between number of nodes and the number of decision variables in the model.

#### IV. METHODS FOR REDUCING COMPLEXITY

As mentioned, the “full” airspace model incorporates every planar airspace location, replicated over all time steps, resulting in too large a model to solve meaningfully sized problems given fixed computation resources. This section identifies a number of heuristic approaches for the improved *construction* of the network model itself, i.e., reduction in the number of nodes, to extend the solution space to larger scales. The optimization model formulated in the previous section can then be solved for each reduced network in less time, or preferably,

<sup>1</sup>All computations were performed on a high-end workstation with 3.5 GHz Intel i7 CPU, overclocked to 4.5 GHz, 32Gb of RAM, 640GB SSD, 3 TB of virtual memory.

the maximum size of the scenario (e.g., the airspace) that can be solved is now larger and approaching operational relevance.

The first three heuristics use geometric considerations (both spatially and temporally) to selectively cut nodes from the network.

*a) Time slice:* Our first attempt to reduce the number of nodes is to eliminate nodes based on when they could be encountered by any UAV, i.e., the reachable set as a function of time. Given maximum speeds of the UAV, straightforward geometric calculations can determine which nodes to include for each time step. In other words, let  $N_{ts}$  as this reduced set of nodes using this heuristic, defined as

$$N_{ts} = \{n \in N \mid \text{dist}(n, n' \in N_0) \leq T\},$$

where  $\text{dist}(i, j)$  represents Manhattan grid distance between nodes  $i$  and  $j$  and corresponds to the reachable nodes (since we assume the UAV can transit to each adjacent node in a single time step).

*b) Convex hull:* This approach uses the convex hull of the landing site locations and the starting positions of the UAVs to define valid nodes for the network. This was done by taking the convex hull in continuous space, and removing all the nodes that did not lie within the hull. The assumption is that the aircraft can navigate freely within the convex hull boundary. The time slice approach is then applied to only include reachable nodes in the network. Thus, we define the set of nodes for this network as

$$N_{cvx} = \{n \in N_{ts} \mid n \in \text{cvx\_hull}(N_0, A)\},$$

where  $\text{cvx\_hull}$  denotes the subroutine which computes the nodes within the convex hull of the set arguments. Recall that determining a convex hull of  $n$  points has an acceptable computational complexity of  $\mathcal{O}(n \log n)$  [15].

*c) Multiple convex hull:* Similar to the Convex hull approach, this heuristic constructs a convex hull of the UAV start locations with each airport individually first, then takes the union of these sets to represent the airspace boundaries. Further refinement by reachable nodes in time is performed. Then we find that

$$N_{mcvx} = \bigcup_{a \in A} \{n \in N_{ts} \mid n \in \text{cvx\_hull}(N_0, a)\}$$

This approach effectively removes nodes unlikely to be transited that exist in-between the likely airways to the landing sites.

The previous heuristic methods, though efficient, rely entirely on geometric cuts of the airspace. Alternatively, we can leverage agent-based algorithms and sampling techniques to instantiate large numbers of particles and endow them with simple rules, e.g., transit towards assigned airports and avoid nodes already containing a particle. The swarm-inspired approaches thus leverage Monte Carlo simulation techniques to inform the construction of the network for the optimization model. For the following heuristics, let  $K$  denote the number of simulation runs of the randomized transits.

*d) Agent-based masks:* Large numbers of particles or agents are randomly instantiated at possible UAV starting locations and provided several rules, including random process noise, to transit to assigned landing sites. Over the course of  $K$  Monte Carlo simulation runs, we then track the visited airspace that the agents occupied in each time step and created a convex hull for occupied airspace by all agents. Then the nodes contained within this network model is the set described by

$$N_{abm} = \bigcup_{t=1}^T \left( \text{cvx\_hull} \left( \bigcup_{k=1}^K N_t^k \right) \right),$$

where  $N_t^k$  represents the set of nodes occupied by the UAVs at time step  $t$  in the  $k^{\text{th}}$  simulation run. This procedure gives us natural time slicing throughout the network, while still providing enough space for the UAVs to move about without significant deconfliction issues.

*e) Organic masks:* Analogous to the Multiple convex hulls heuristic, this approach separately tracks the set of particles assigned to each landing site over  $K$  Monte Carlo simulation trials, constructs their respective convex hulls, and then composes the resulting sets. In this case, we have the reduced set of nodes defined as

$$N_{org} = \bigcup_{a \in A} \left( \bigcup_{t=1}^T \left( \text{cvx\_hull} \left( \bigcup_{k=1}^K N_{a,t}^k \right) \right) \right),$$

where now  $N_{a,t}^k$  is the subset of nodes at time step  $t$  in simulation run  $k$  occupied by UAVs assigned to airport  $a$ . As with the previous agent-based mask, this approach further captures the probabilistic locations of UAVs as they transit (subject to random perturbations and deconfliction tie-breaks) to their assigned landing sites.

For a fixed airspace size (e.g.,  $60 \times 60$ ) for which the full airspace (i.e., no reduction in nodes) is still (barely) solvable, Figure 3(a)-(f) illustrate the general shapes of the resulting networks for the Full, Time slice, Convex hull, Multiple convex hull, Agent-based masks, and Organic masks heuristics in the context of the reference scenario (see Figure 1), with their respective number of nodes tabulated below.

$ N $	$ N_{ts} $	$ N_{cvx} $	$ N_{mcvx} $	$ N_{abm} $	$ N_{org} $
356,430	97,230	42,366	23,628	40,437	19,123

As seen, the significant reduction in size can then be leveraged to solve the optimization model for larger problems. Presentation of results is deferred to Section V.

## V. SIMULATION STUDIES

In this section, we present additional reference scenarios as part of the experimental design to highlight the computational trade offs of the proposed network node cutting heuristics. The solution to the network-based model using the full airspace is taken to be the true optimal solution, since it contains all of the nodes and edges that can possibly be included. We compare all algorithms against this benchmark optimal solution.<sup>2</sup>

<sup>2</sup>MATLAB source for the presented simulation and algorithms is available at <http://faculty.nps.edu/thchung>.

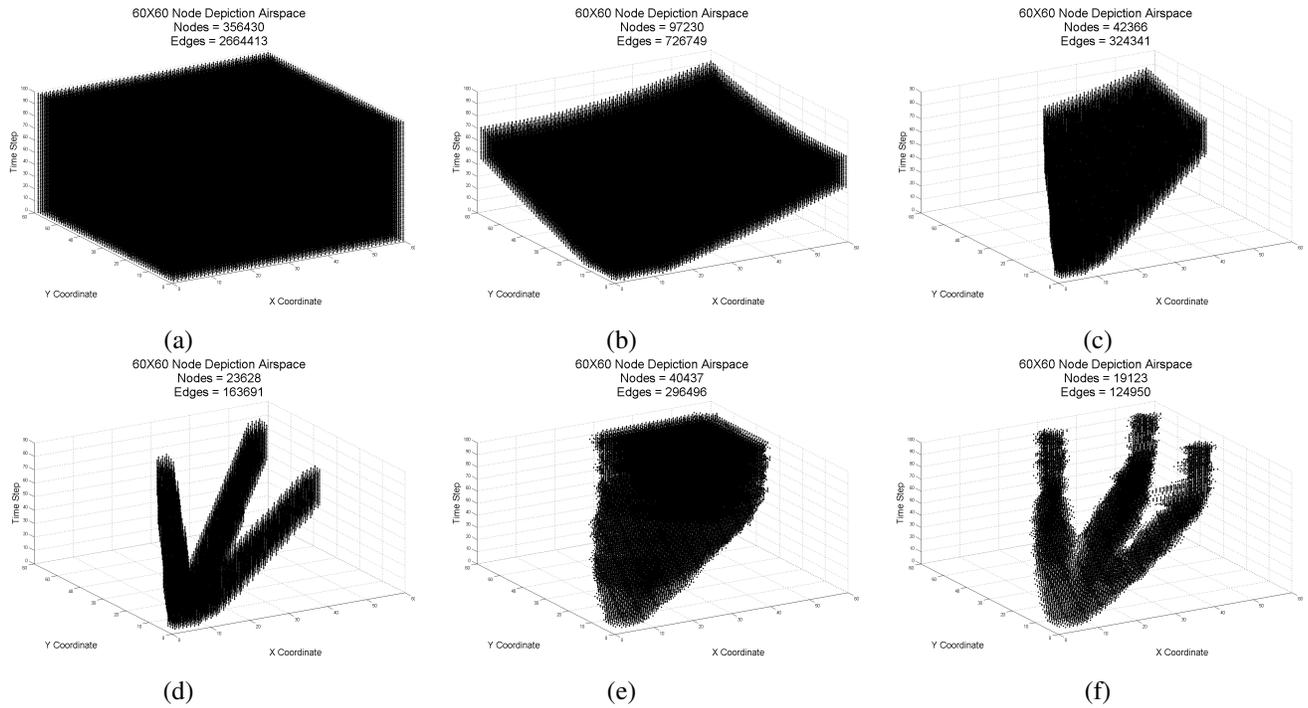


Fig. 3. Illustration of the time-expanded networks for various network constructions (with number of nodes in brackets): (a) Full airspace [356,430]; (b) Time slice [97,230]; (c) Convex hull [42,366]; (d) Multiple convex hull [23,628]; (e) Agent-based masks [40,437]; (f) Organic masks [19,123]

The five reference scenarios are described below and, in addition to Figure 1, are illustrated in Figure 4(a)-(d). For each scenario, other than landing site configurations, other relevant parameters are held constant: number of UAVs is 25, airspace size is  $60 \times 60$ , and the landing interval at all airports is one time step.

- 1) **Scenario 1 - Close Landing Zones:** This is what we have explored thus far in the previous sections as the baseline scenario. In this instance we have a small group of three landing zones in relatively close proximity (i.e., they reside in the same quadrant).
- 2) **Scenario 2 - Dispersed Landing Zones:** Scenario 2 has three landing zones similar to Scenario One; however, the landing zones are now placed in the three corners opposite the UAVs' starting positions.
- 3) **Scenario 3 - Overflight of Landing Zone:** This scenario now has four landing zones, but one of the landing zones is directly in the path of another. In other words, two UAV groups for these coinciding paths now must transit and deconflict in the same restricted airspace for the two different airports. This feature is to observe the impact of a more challenging deconfliction setting on the different algorithms, and see if there is an effect on the ability to find an optimal solution.
- 4) **Scenario 4 - Airports in Each Corner:** This scenario has four landing zones, such that the UAVs are initially located towards the center of the airspace and the landing zones are located at each distant corner of the airspace. This experiment allows all of the airports to be equidistant from the starting position of the UAVs and from each

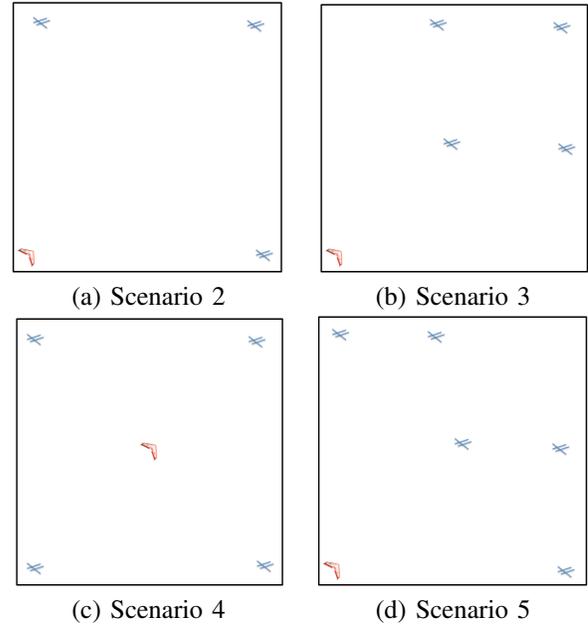


Fig. 4. Graphical Depiction of Reference Scenarios. Simulation results vary these scenarios, i.e., different landing site configurations and numbers, while holding other scenario parameters fixed, i.e., number of UAVs is 25, airspace size of  $60 \times 60$ , and a landing interval of one time step.

other, which maximizes their spatial dispersion and may lead to an improvement in performance due to earlier segregation of the UAVs.

- 5) **Scenario 5 - Five Airports at Multiple Distances:** This scenario introduces five landing zones that are

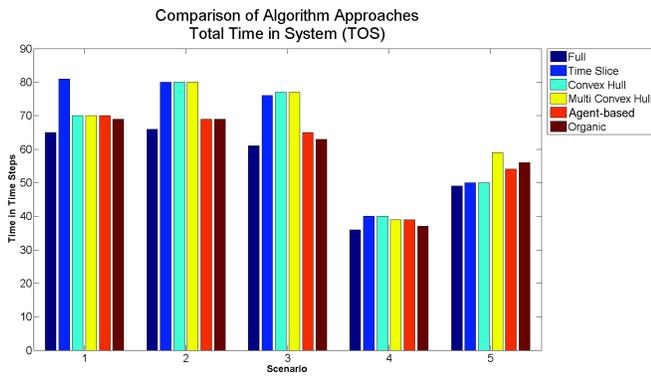


Fig. 5. Graph of the mission performance (objective function) to minimize total time in system, i.e., total time until all UAVs are landed.

at different distances. This more challenging setting explores the node selection capability of the algorithms with many landing zones at a variety of distances. The scenario also tests the network-based model’s ability to assign UAVs to five landing zones. As such, this scenario provides the most difficulty from both the set-up and the computational solution perspectives.

Across these different scenarios, we perform extensive numerical studies and assess the performance for the various network construction approaches using various metrics, including the mission objective of minimal time in system, but also computational measures of the number of nodes, amount of memory required for computations, and the runtime of the optimization. In particular, we can see in Figure 5 that, as expected, the objective function is minimal for all scenarios when using the Full network compared to other reduced node sets. Notably, the mission performance of the sampling-based approaches (that is, the agent-based and organic heuristics) are comparable in most of the scenarios (with the exception of Scenario 5), which highlights their value, depending on the acceptability of slightly less than optimal solutions. The geometric approaches, such as the convex hull and multiple convex hull heuristics, do not yield as good solutions. For example, in the case of the multiple convex hull algorithm, the resulting measure for the total time in system can be approximately 30% greater than the Full solution, which in cases of urgency may be unacceptable.

In addition to mission performance, however, of interest in this paper is the computational tractability of these otherwise large swarm transit and landing problems, and so the computational metrics are further investigated and illustrated in Figures 6-9. Though the Full network ensures that the generated solution minimizes the total time in system, the problem becomes too large for efficient computation as the airspace and the number of UAVs increases. The multiple convex hull and organic algorithms have the most node cutting power (see Figure 6), and are able to reduce the computation enough to make larger problems more tractable. Even with the convex hull heuristic, for example, solutions can be found for significantly larger problems (previously untenable for the

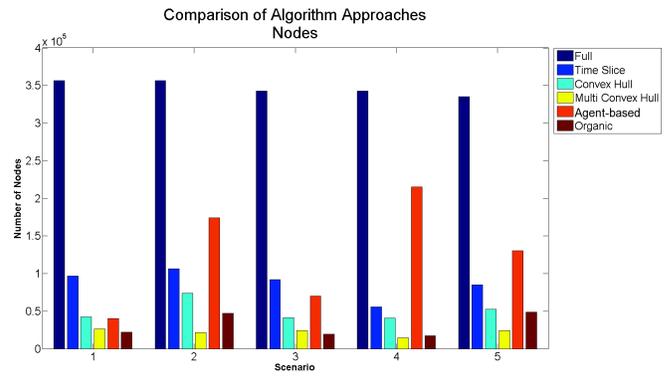


Fig. 6. This graph shows the significant node reduction capability of the algorithms with respect to the Full airspace model, with the organic and multi-convex hull algorithms reducing the number of nodes most significantly.

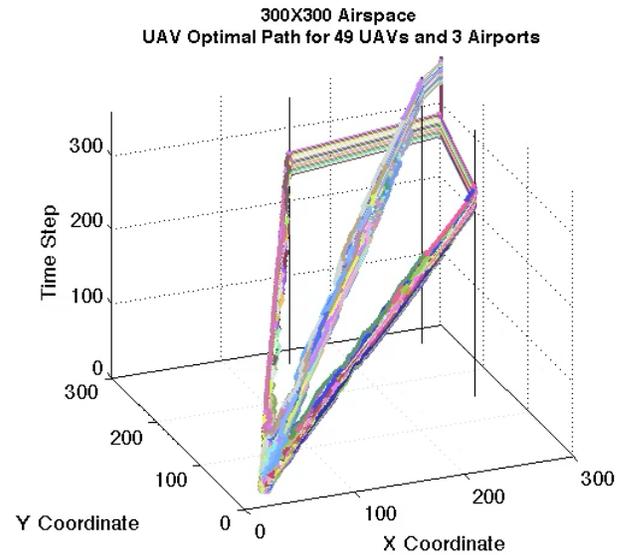


Fig. 7. Illustration of transit trajectory evolutions for Scenario 1 for significantly larger problem scope involving an airspace size of  $300 \times 300$  and 49 UAVs using the convex hull heuristic for reducing the number of nodes in the network. Each line represents an individual UAV’s path through the airspace; once the UAV reaches the airport it is shown traveling to an arbitrary point (representing the abstract sink node).

Full network), such as illustrated in Figure 7, with an airspace size of  $300 \times 300$  (versus  $60 \times 60$ ) and 49 UAVs (instead of 25 UAVs).

In terms of memory resources, however, the least expensive approach can be seen in Figure 8 to consistently and decidedly be the multi convex hull method. In cases of limited computational capabilities, often exhibited in field deployed systems, this approach may be the most favorable for generating good solutions, in contrast to its more costly alternatives.

Another metric that can be used to help select which approach should be used is the computational runtime (measured in seconds), which may be relevant for time-sensitive generation of feasible (if not optimal) solutions. As shown in Figure 9, the presented heuristics offer significant savings in runtime, generally offering solutions for nearly all scenarios

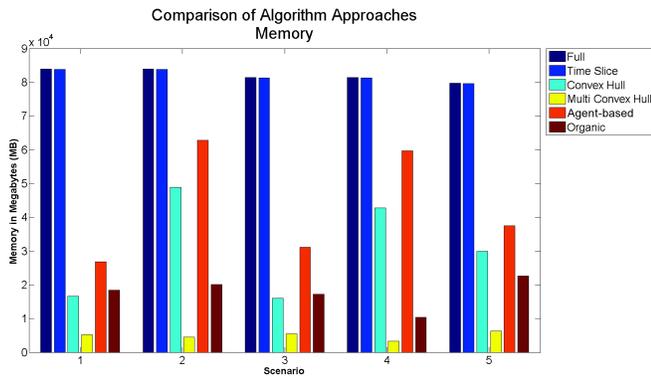


Fig. 8. Comparison of the amount of memory used by GAMS/CPLEX to solve each algorithm, as measured in megabytes (Mb).

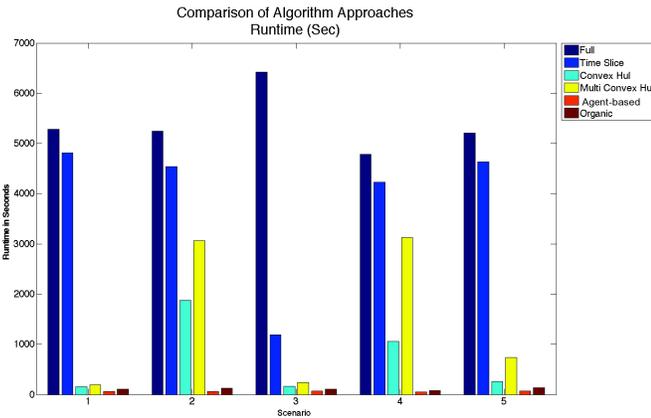


Fig. 9. The execution time, as measured in seconds, highlights the variable amounts of time to find a solution for the different algorithms in GAMS. Once again, we find the multi-convex hull and organic mask algorithms to (nearly identically) find the solutions fastest across all scenarios

in less than three minutes. Once again, the simulation-based heuristics outperform their geometric counterparts in reducing runtimes, and demonstrate their usefulness across different landing site configurations.

However, an interesting result of the sampling-based approaches can be witnessed in certain configurations and scenarios. Consider the organic mask heuristic, which yields transit and landing trajectories illustrated in Figure 10 and a resulting objective value that is relatively near the Full network’s benchmark values. However, in many of the simulation studies, the organic mask approach violates the mission constraint of the minimum number of UAVs at a given airport (c.f. Constraint 4). In other words, a feasible solution arises in which fewer UAVs land at a given landing zone than is required. For example, in reference Scenario 1 with three airports, the constraint of eight UAVs per landing site is specified in the network optimization model. Nevertheless, as seen in the landing sequence diagram shown in Figure 11, one airport receives only seven UAVs (with ten and eight at the other two). This requirement constraint is consistently broken in a number of test scenarios explored. This violation appears to occur because the nodes remaining after applying

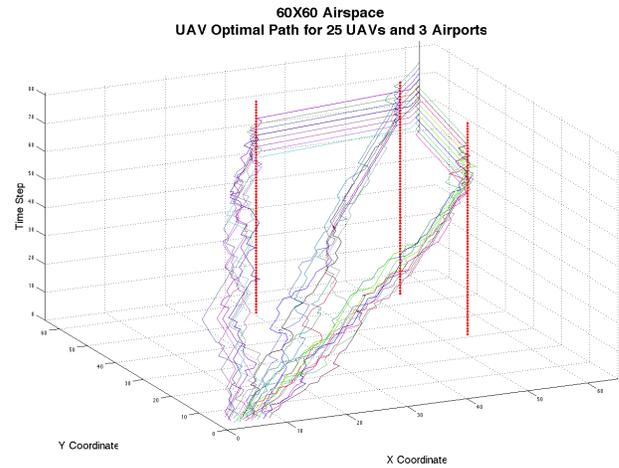


Fig. 10. Optimal path of UAVs in  $60 \times 60$  airspace using the Organic mask heuristic. The UAVs start to spread out early in the airspace at their position for landing. Each line represents an individual UAV’s path through the airspace; once the UAV reaches the airport it is shown traveling to an arbitrary point (representing the abstract sink node).

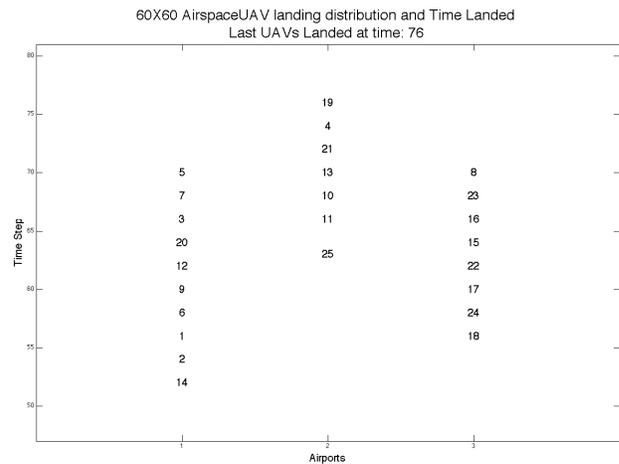


Fig. 11. Depiction of landing sequence of UAVs at three landing sites using the Organic mask approach. Notice how Airport 2 has only seven UAVs land there, where the other two landing zones have ten and eight, although each landing zone is supposed to have at least eight per. This is an example of the organic mask algorithm relaxing the “required” constraint.

the organic mask heuristic do not offer enough “maneuver room” for the UAVs; the closest landing zone takes more UAVs than the other landing zones to alleviate this space restriction.

Thus, the ever-present tradespace of computation versus optimality is highlighted in these studies. Ultimately, the decision maker employing these swarms of UAVs must assess and appropriately select which approach is best suited, given the priorities or constraints on the mission. Based on the exploration provided in this paper, the multiple convex hull approach offers one of many promising avenues for further investigation into larger airspace examples and comparison with additional simulation models. The multiple convex hull approach consistently yields nearly optimal solutions, meets

all mission constraints, and offers a relatively computationally cheap approach across diverse mission scenarios.

## VI. CONCLUSIONS AND FUTURE WORK

The melding of both simulation and optimization can produce results that are both tractable and more consistently closer to optimal than simulation alone. Techniques such as agent-based simulation can be used in traditional optimization models to gain insight into systems and to give the modeler a localized area for finding the solution. In this article, we use agent based models to make otherwise computationally intractable problems tractable, while still maintaining a near optimal solution.

Similarly, formulation as an optimization problem gives the simulation modeler analytic bounds on the behavior that is desired, and a benchmark for the measure of performance for a simulation. This benchmark is important for transitioning or extrapolating insights attained from simulation to real and operational systems

Some of the techniques developed can be used, not only for the paths of individual UAVs, but for providing a basis for the use of heuristic and exact optimization solution methodologies together. In this example of UAV swarms, a network-based model can be formulated and used for routing and landing of large UAV teams, and heuristic models can be used for the micro-level control of collision avoidance and flocking, thereby providing a more defined balance between distributed and centralized control.

This work further provides important insight into the level of command and control that is required in swarming systems. In general, limitations of agent-based models include suboptimal collective performance, since the agents are not provided with a common “big picture” understanding on how to achieve their goal; they act purely on local and immediate information. The use of agents in the network-based model produced results that were suboptimal but near the benchmark optimal solution. As such, network optimization models should be used for higher level optimization tasks, for example, landing zone assignment, general transit routing, and large scale airspace deconfliction, relegating the low level control for collision avoidance and flocking to the individual UAV. The most efficient use of resources, both computationally and temporally, may be to have a combination of centralized and distributed optimization and control approaches.

Insights from this study identify a number of avenues for future work. The time-expanded formulation straight forwardly enables investigation of *moving* landing sites, given their prediction motion models. This extension provides relevance to applications of target-tracking, surveillance, and task assignment leveraging the network-based optimization model formulation. The relationship between the centralized network optimization problem and the decentralized stability and robustness of flocking algorithms [16], graph-theoretic coordination algorithms [17], and other distributed multi-agent algorithms remains an active open area of research. Further, as realistic dynamics of the aerial platforms was not addressed

in this work, current research efforts leveraging physics-based modeling (e.g., constant curvature turns) and virtual environments (e.g., flight simulations including environment disturbances) are ongoing. These continuing efforts are precursors to upcoming live-fly field experiments with a fleet of unmanned aerial vehicles at Camp Roberts, Calif. utilizing both agent-based and network optimized algorithms in real operational settings.

## REFERENCES

- [1] W. D. Shannon, “Future Trends in Unmanned Aviation and Weapons,” 2012.
- [2] M. D. Peterson, D. J. Bertsimas, and A. R. Odoni, “Decomposition Algorithms for Analyzing Transient Phenomena in Multiclass Queuing Networks in Air Transportation,” *Operations Research*, vol. 43, no. 6, pp. 995–1011, Nov. 1995.
- [3] M. A. Bolender and G. L. Slater, “Evaluation of scheduling methods for multiple runways,” *Journal of Aircraft*, vol. 37, no. 3, pp. 410–416, Oct. 2000.
- [4] A. P. Saraf and G. L. Slater, “Optimal Dynamic Scheduling of Aircraft Arrivals at Congested Airports,” *Jrnl of Guidance, Control, & Dynamics*, vol. 31, no. 1, pp. 53–65, Jan. 2008.
- [5] J. R. Boesel, “An Air Traffic Simulation Model that Predicts and Prevents Excess Eemand,” The MITRE Corporation: Center for Advanced System Development (CAASD), McLean, Virginia 22102, Tech. Rep., 2003.
- [6] P. Brooker, “Simple Models for Airport Delays During Transition to a Trajectory-Based Air Traffic System,” *Journal of Navigation*, vol. 62, no. 04, p. 555, Oct. 2009.
- [7] P. Dell’Olmio, “A New Hierarchical Architecture for Air Traffic Management: Optimisation of Airway Capacity in a Free Flight Scenario,” *European Journal of Operational Research*, vol. 144, no. 1, pp. 179–193, Jan. 2003.
- [8] K. Artiouchine, P. Baptiste, and J. Mattioli, “The K King Problem, an Abstract Model for Computing Aircraft Landing Trajectories: On Modeling a Dynamic Hybrid System with Constraints,” *INFORMS Journal on Computing*, vol. 20, no. 2, pp. 222–233, Sep. 2007.
- [9] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” *ACM SIGGRAPH Computer Graphics*, vol. 21, no. 4, pp. 25–34, Aug. 1987.
- [10] S. R. Conway, “An Agent-Based Model for Analyzing Control Policies and the Dynamic Service-Time Performance of a Capacity-Constrained Air Traffic Management Facility.” ICAS, 2006, pp. 1–8.
- [11] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 401–420, Mar. 2006.
- [12] H. Yu, J. Jian, and S. Yanjun, “Flocking Control of a Group of Agents Using a Fuzzy-Logic-Based Attractive/Repulsive Function,” *Int’l J. of Communications, Network and System Sciences*, vol. 03, no. 06, pp. 569–577, 2010.
- [13] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, 1993.
- [14] *CPLEX 12*, GAMS Development Corporation, 2012. [Online]. Available: <http://www.gams.com/dd/docs/solvers/cplex.pdf>
- [15] M. de Berg, O. Schwarzkopf, M. van Kreveld, and M. Overmars, *Computational Geometry*, 3rd ed. New York, New York: Springer, 2000.
- [16] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.
- [17] F. Bullo, J. Cortés, and S. Martinez, *Distributed Control of Robotic Networks*. Princeton, 2009.