Evolving Dynamically Reconfiguring UAV-hosted Mesh Networks

Rahul Dubey, Sushil J Louis, Shamik Sengupta
Computer Science and Engineering
University of Nevada
Reno, USA
rdubey018@nevada.unr.edu, sushil@unr.edu, ssengupta@unr.edu

***Abstract*—We use potential fields tuned by genetic algorithms to dynamically reconfigure unmanned aerial vehicles networks to serve user bandwidth needs. Such flying network base stations have applications in the many domains needing quick temporary networked communications capabilities such as search and rescue in remote areas and security and defense in overwatch and scouting. Starting with an initial deployment that covers an area and discovers how users are distributed across this area of interest, tuned potential fields specify subsequent movement. A genetic algorithm tunes potential field parameters to reposition UAVs to create and maintain a mesh network that maximizes user bandwidth coverage and network lifetime. Results show that our evolutionary adaptive network deployment algorithm outperforms the current state of the art by better repositioning the unmanned aerial vehicles to provide longer coverage lifetimes while serving bandwidth requirements. The parameters found by the genetic algorithm on four training scenarios with different user distributions lead to better performance than achieved by the state of the art. Furthermore, these parameters also lead to superior performance in three never before seen scenarios indicating that our algorithm finds parameter values that generalize to new scenarios with different user distributions.**

***Index Terms*—UAV network, distributed control, genetic algorithms, potential fields**

I. Introduction

Unmanned Aerial Vehicle (UAV) networks have many potential applications in emergency, defense, and security applications where there is a need to quickly deploy a communication network. Flying UAVs can serve as base stations for civilian and military uses such as search, rescue, surveillance, patrolling, and mapping, especially in difficult, remote terrain [1]. However, many challenges remain before large number of UAVs can be efficiently and effectively deployed. For example, in a search and rescue scenario we have to first distribute available UAVs over an area of interest so that we can cover the entire area, then dynamically continue repositioning UAVs to efficiently (save battery life) serve found users. Repositioning enables maintaining connectivity to a fixed operations Command Center (CC) while serving moving users and changing bandwidth needs [2].

There are only some of the many issues in creating and maintaining a mesh network. This paper focuses on the problem of maneuvering a large number of UAVs that create and maintain a mesh network that connects all users to the command center for the maximum amount of time. That is, a network that maximizes bandwidth coverage while maximizing UAV flying time.

UAVs network deployment algorithms can be centralized or distributed [3]. Centralized control algorithms work well in well understood environments; typically not found in emergency situations. However, even in well understood environments, long communication delay caused by multiple hops and limited bandwidth may prevent centralized control algorithms from being used in time delay sensitive areas and remote exploring applications. We therefore investigate distributed control algorithms based on using potential fields to dynamically reconfigure a UAV network in relatively unknown environments.

A. Problem formulation

Flying UAV nodes implement a mesh network that maximizes bandwidth coverage and flying time (network longevity) while maintaining communications back to a command center. Optimal, or perfect, bandwidth coverage means providing every user on the ground with required bandwidth. Each user may need text, voice, video or a combination of these services each with different bandwidth requirements. This has consequences for network longevity. Assuming UAVs use the same amount of power for hovering and moving which depletes the battery at a constant rate, then in our problem we need only consider that battery usage is proportional to served bandwidth. The less bandwidth a UAV serves, the less battery is used and the longer it can stay aloft. Thus, we assume that given a fixed number of UAVs, the more UAVs we can recruit to serve a required bandwidth, the longer the network lasts.

We assume users are non-uniformly distributed over the area of interest and that UAVs have user locating sensors that sense users within U_{gr} distance. Furthermore, UAVs can communicate with each other at a maximum range of U_{ar}. For a user to communicate with the command center, we will then need a chain of UAVs at most U_{ar} apart all the way from the user to the command center so that data hops from one UAV node to the next from the user to the command center and vice-versa. Figure 1 shows two clusters of users (16) covered by five (5) UAVs. Since UAVs also need to communicate in order to move to avoid collisions and optimize the network, this communication will also need bandwidth. To minimize

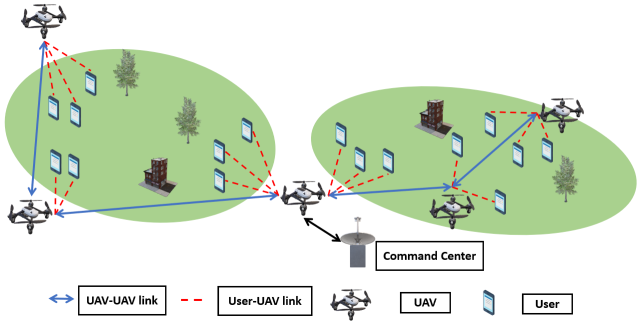


Fig. 1: An example of UAVs network deployment.

such communication, we assume that UAVs only communicate with their closest (one-hop) neighbors.

In our scenarios, UAVs start at the command center in the center of a rectangular Area Of Interest (AOI) and are deployed to cover the AOI in order to locate all users while maintaining a mesh network connected to the CC. Once this phase completes, UAVs move to provide coverage and maximize lifetime over the non-uniformly distributed users in the AOI. Given these assumptions, we use a genetic algorithm to tune the parameters of potential fields to govern the movement of UAVs. Since the number of available UAVs determines coverage and lifetimes, we also consider scenarios with different numbers of available UAVs.

Early experimental results show that the genetic algorithm tuned potential fields based algorithm outperforms the state of the art Adaptive Triangular Deployment Algorithm (ATRI) on four training and three test scenarios each with different numbers of UAVs and different numbers and distributions of users on the ground. These experiments do not consider battery usage for moving since ATRI and other prior approaches do not consider movement and only provide desired positions to move to - not how to get to the desired positions. With potential fields, collision free movement to desired locations naturally emerges with no additional work. Furthermore, our proposed algorithm improves network lifetime compared to ATRI and serves more data to users on the ground over network lifetime.

The remainder of this paper is organized as follows. The next section describes prior work in mesh networks and potential fields based group movement. Section III specifies our training scenarios, and movement model, movement. Next, section III-A describes our algorithm, how potential fields determine UAV movement, our fitness function, and provides a brief overview of the ATRI algorithm. Our experimental setup and results constitute section IV. The last section provides conclusions and future work.

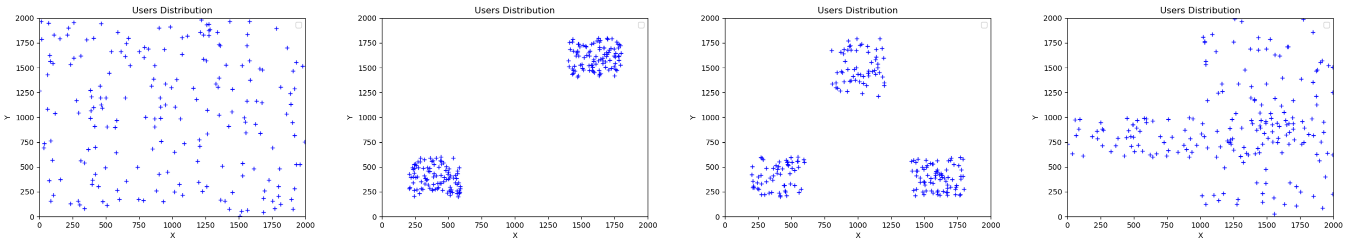
II. Related Work

UAV deployment for on demand wireless coverage has several real world applications and has been attracting increasing interest [1]. In terms of control, we focus on distributed control since that is most closely related to the work in this paper [3]. Mesh network creation and maintenance presents multiple challenges when using UAVs as base stations. Minimizing deployment time, placing UAVs optimally to increase wireless coverage, minimizing number of UAVs to be deployed, maintaining a mesh network connected to ground base station, and extending lifetime of deployed UAVs, routing, channel allocation, all present significant challenges.

To minimize deployment delay, Zhang presented fast deployment algorithms in [4]. Fast deployment of UAVs as flying base stations during emergency situations may help to save people trapped in a wildfire in a forest or in other natural disasters. In [5], authors presented a dynamic algorithm to serve a sub-region within the AOI that requires more bandwidth. The study assumed an uniform distribution of users while we look at non-uniform distributions of users.

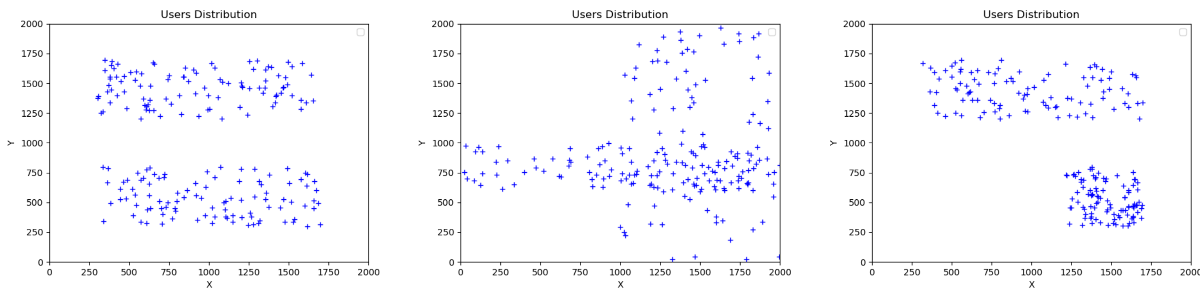
The complexity of UAV deployment over AOI increases with unknown user distributions and would be more common in emergency or first responder situations. When searching for users in need of rescue, we want to blanket the AOI and one possible way to do so is to distribute UAVs in a covering pattern over the AOI. Common methods for such deployment use the circle point theorem [6] and adaptive triangular deployment (ATRI) based on delauney triangulation [7]. Using one hop neighbor state information and bandwidth demand, ATRI generates a mesh network within a given AOI and to the best of our knowledge, seems to be the state of the art for UAVs network deployment. In [8] Lyu, proposed a placement optimization technique to minimize number of unmanned aerial vehicle-mounted mobile base station while proving wireless coverage to ground terminals. The algorithm works in polynomial time with successive mobile base station placement. In [9], authors initially deploy UAVs using Circle Packing Theory [6] and then proposed an approach to adjust the altitude of deployed UAVs to increase and decrease an UAV's sensing area. Using this approach, the bandwidth area coverage is maximized but as the altitude of deployment increases, the signal strength reduces resulting in poor coverage quality.

Potential field controllers have been used to guide autonomous agents in many different application domains. La in [10] uses a potential field based controller for dynamic tracking of a target while avoiding obstacles. Louis and others have used potential fields for 2D and 3D distributed autonomy in games [11]–[14]. Zhao presented a centralized and a distributed algorithm for UAVs deployment while maintaining connectivity among UAVs [3]. However, they considered only two different types of user distributions - uniformly random and in three clusters spread around the AOI with a command center in the middle. In this work, we build on the idea and investigate several different user distribution scenarios using an evolutionary distributed control algorithm to maximize both bandwidth coverage and longevity of the network. Our approach uses potential fields to guide UAVs in unknown environment and the GA evolves the parameters of potential fields. The next subsection explains our research environment and methodology for experiments.



(a) S1 (b) S2 (c) S3 (d) S4

Fig. 2: The four training scenarios used for fitness evaluation. Note that each scenario has a different user (blue +) distribution.



(a) S5 (b) S6 (c) S7

Fig. 3: The three testing scenarios distinct from training scenarios and never seen during parameter evolution.

III. Methodology

For simulations, we used Unity3D software to create users, UAVs and different scenarios. We use a $(\lambda+\mu)$ evolutionary algorithm with simple crossover and mutation of binary encoded chromosomes. The binary encoding enabled easy control of search space size. Evaluating fitness comprised four steps.

1) Send chromosome of potential field parameters to UAV net simulation
2) Run simulation for n time steps and gather variable values
3) Compute fitness from variable values
4) Return fitness

Prior work and our own experience in initial testing indicated that using one scenario (one user distribution and a set number of UAVs) during fitness evaluation often resulted in lower performance in other scenarios not seen during evolution. We thus generated and used four scenarios during evaluation. That is, one fitness evaluation repeated the four steps above for four different distributions of users and returned the average of the fitness over the four scenarios. Figure 2 shows these *training* scenarios. From left to right, we have an uniform distribution of users throughout the AOI, users uniformly distributed among two clusters, and users uniformly distributed among three clusters. The rightmost scenario uniformly distributes users along a horizontal central band and along the right side of the AOI.

Once our evolutionary algorithm was done, we took the best individual from the last generation and evaluated this individual's fitness on three different *testing* scenarios. These testing scenarios are show in Figure 3. From left to right, users are distributed on two horizontal band in upper and lower side of AOI, users distribution similar to 4^{th} training scenario but with different concentration of users, and lastly, in the 3^{rd} test scenario users are in two clusters with different distributions. Doing well on never before seen test scenarios provides evidence supporting our approach's robustness. We note also, that the computationally intensive fitness evaluation simulation took several seconds to run thus restricting us to relatively low population sizes.

A. Representation and Computing Fitness

In this work, we assume that the AOI is a square of length 2000 meters and that the command center is located in the middle at $(1000, 1000)$. All UAVs start at the center within a $10x10$ meter area and we assume that they maintain a constant altitude of 100 meters. Table III lists these and other simulation parameters and genetic algorithm parameters.

The UAVs move in two phases. First, to find users, UAVs spread out to cover the AOI. Active UAVs (AUs) start providing coverage for users within their service radius, U_{gr}. UAVs not within range of any users are called Inactive UAVs (IUs). The first phase runs for 1500 time steps and user positions and final UAV positions are shown in Figure 4. This phase uses one set of hand-tuned potential fields parameters to move. In the figure, red dots represent users, red lines connect users to their server UAV and green lines indicate neighboring UAVs.

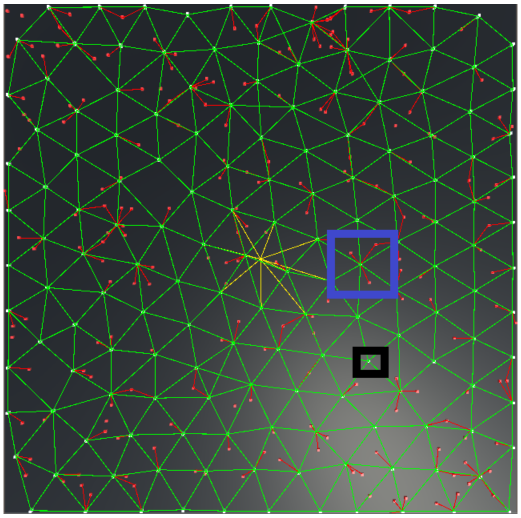


Fig. 4: Initial UAV deployment over the area of interest. Red dots represent users, lines represent connections between UAVS (green) and User-UAVs (red).

The blue rectangle shows an active UAV serving 3 users while the black rectangle indicates an inactive UAV serving no one.

The second phase runs Algorithm 1, our proposed algorithm called Evolutionary Adaptive Network Deployment Algorithm ($EANet$). Algorithm 1 deploys UAVs over a given AOI by computing potential fields and then uses the final UAV locations to compute the deployment's fitness. This phase uses a second set of potential fields specified by the genetic algorithm to move. The outer loop, loops over the four training scenarios and runs each scenario for $MaxTimeSteps$. At each time step, the `AssociateUsers` function in Algorithm 1 finds users within sensor range (U_{gr}) and gathers data tracking user coverage and bandwidth consumption. For each UAV, `FindNeighbors` finds other UAVs within Air-to-Air (A-2-A) communication range (U_{ar}) and records them as neighbors. Once each UAV knows information about its associated users and neighbor UAVs, `ComputePotentialFields` computes heading from the potentials fields at its current location. `moveAll(Heading)` then moves each UAV in the direction of this heading at maximum speed. At $MaxTimeSteps$, the algorithm computes bandwidth coverage provided by UAVs to users in the AOI using `findBQCoverage` and the total number of active UAVs computed by `FindActiveUAVs`. The algorithm then computes a normalized fitness between 0 and 1 for each of bandwidth coverage and active UAVs using $MaxBW$, the maximum bandwidth coverage demanded by users, and $MaxUAV$, the total number of UAVs available for deployment. Summing these two factors results in a fitness between 0 and 2 for each scenario. At the end of the for-loop, the algorithm returns average fitness over $MaxScenarios$. Note again that the genetic algorithm only tunes the potential fields for this second phase.

Algorithm 1: EANet deployment and fitness computation

```
Input  : Initial position of UAVs, AOI, Candidate
         Solution
Output: fitness
fitness = 0;
MaxScenarios = 4;
MaxTimeSteps = 1500;
for scenario in MaxScenarios do
    timeSteps = 0;
    while timeSteps<MaxTimeSteps do
        AssociateUsers(scenario);
        FindNeighbors();
        Headings = ComputePotentialFields();
        MoveAll(Headings);
        timeSteps++;
    end
    bwCoverage = FindBQCoverage();
    AUs = FindActiveUAVs();
    fitness += bwCoverage / MaxBW + AUs /
      MaxUAV;
end
fitness = fitness / MaxScenarios;
return(fitness);
```

B. Potential fields

Potential fields have been used extensively in robotics and games for fast, real-time, collision free movement [12], [15], [16]. Each potential field of the form cd^e where d is distance has two tunable parameters (c, e) that determine field effect. During the first phase, when the relatively simple goal is to maximize coverage of the AOI while maintaining the mesh network, we hand tune these parameters. We can use multiple other techniques that work equally well on this simple first phase [6], [7]. Since we start at the center, we only need a single repulsive potential field to achieve desired coverage as shown in Figure 4. We run this phase for 1500 simulation steps, sufficient to cover the area. Once UAVs are deployed, each UAV associate users within a sensing range of U_{gr} and finds neighboring UAVs in the range U_{ar}.

Figure 5 shows user and UAV distribution at the end of the second phase. Equation 1 shows the four potential fields that affect a specific UAV's (UAV*) movement during the 1500 step second phase. The vector sum of these potential fields given by PF_{uav} provides a desired heading for the UAVs to turn to while moving at a speed. Specifically UAV* moves in the direction given by PF^*_{uav}'s direction vector.

$$\vec{PF^*_{uav}} = \vec{PF_{bw}} + \vec{PF_{nbw}} + \vec{PF_{rd}} + \vec{PF_{ac}} \qquad (1)$$

We next described each term on the right hand side. First, PF_{bw} specifies an attractive potential field based on user bandwidth demands and user location. This field attracts UAVs towards high bandwidth areas. Second, since the more UAVs that share a user's bandwidth requirement, the less power

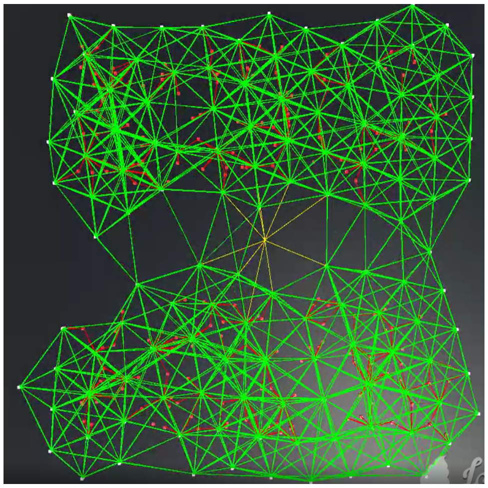


Fig. 5: Final deployed UAV positions on test scenario S5.

used and longer network lifetimes gained, PF_{nbw} specifies a potential field that attracts one-hop UAV neighbors. Third, too much attraction can cause collisions, thus a repulsive distance based potential field, PF_{rd}, repulses UAVs from each other and avoids collisions. Finally, the command center attracts UAVs in order for the mesh network to be able to connect the CC. A potential field specified by PF_{ac} that *only* acts on inactive UAVs, takes care of this requirement. Noting that each term in the equation above (Equation 2) is itself a vector sum over all considered UAVs, we can rewrite the equation as follows.

$$\begin{aligned} \vec{PF}^k_{uav} &= \sum_{i=0}^{n} c_1 bwd_i^{e_1} + \sum_{i=0}^{m} c_2 bws_i^{e_2} \\ &+ \sum_{i=0}^{m} c_3 d_i^{e_3} + c_4 d_i^{e_4} \end{aligned} \tag{2}$$

Here n, the upper limit on the first term is the number of users and the vector is directed from the UAV* towards the i^{th} user with bandwidth demanded, bwd, as magnitude. m is the number of UAVs and for each element of the second term, the vector is directed from the i^{th} UAV towards UAV* with bandwidth served, bws, as magnitude (if the i^{th} UAV is within one hop). The third term vectors direct away from UAV* towards the i^{th} UAV with distance, d, as magnitude (simply the vector difference in position), and the fourth term's vectors direct towards the command center with distance as magnitude. Tuning the coefficients, cs, and exponents, es of these highly non-linear equations to maximize bandwidth coverage and lifetime served is difficult and we thus use an evolutionary algorithm to do so.

In addition to the four coefficients and exponents, we use one additional parameter to specify a minimum distance threshold, $d*$. When computing PF_{rd}, we only count UAVs within $d*$. Different values of $d*$ have differing impacts on performance . For example, if $d*$ is large then UAVs will not congregate sufficiently due to repulsion from UAVs relatively far away and thus may not share bandwidth demand. Whereas if minimum distance value is small, then UAVs may converge too close leaving some other users without any service.

We encoded these parameters in a binary chromosome where each coefficient is encoded in 14 bits, each exponent in 10 bits, and minimum distance in 8 bits. This encoding enables good control over search space size and makes it easy to restrict and change the ranges of each parameter. The length of our chromosome is thus $14 \times 4 + 10 \times 4 + 8 = 104$ bits giving us a highly non-linear search space size of 2^{104}. Coefficient bounds are $-8192 \leq c_i \leq 8192$ with a precision of 1. Exponents range between $-5.12 \leq e_i \leq 5.12$ with precision of 0.01, $d*$ ranges between $0 \leq d^* \leq 2^8$.

Our EA thus tunes nine parameters to maximize equation 3 to promote candidate solutions that maintain a mesh network connected to the command center while maximizing the total number of active UAVs in the deployed network.

$$\frac{\sum_{i \in MU} bw_i}{MaxBW} + \frac{NumAU}{m} \tag{3}$$

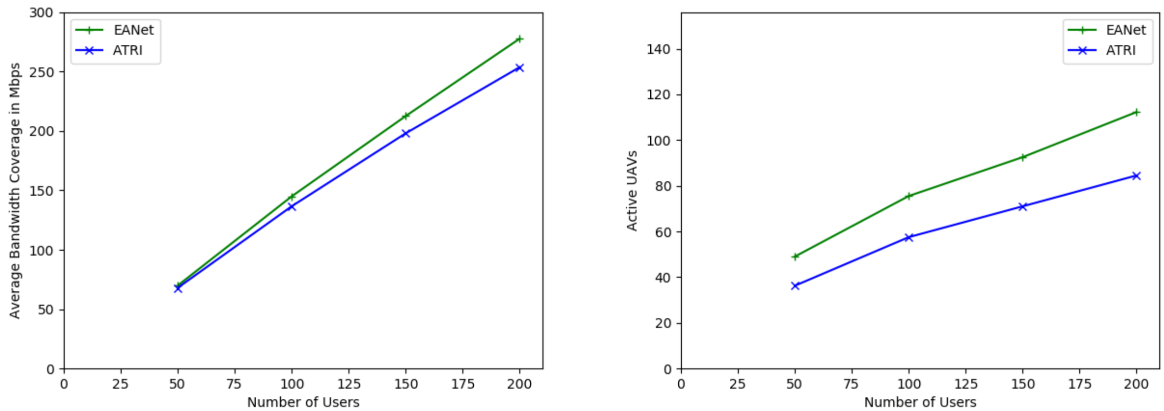
The first term counts the fraction of bandwidth served in network UAVs and the second term counts the fraction of active UAVs. Assuming we have enough UAVs, this function achieves a maximum when all UAVs are serving total bandwidth requirements. Here, MU is the set of all UAVs participating in the mesh network, bw_i is the bandwidth served by the i^{th} UAV. Total request bandwidth over all users is $MaxBW$. In the second term m indicates the number of UAVs available and $NumAU$ is the number of active UAVs. We compare deployed network performance using proposed EANet Algorithm when tuned by our EA against the performance of ATRI, the current state of the art deployment algorithm.

C. Adaptive Triangular Deployment (ATRI)

Adaptive triangular deployment is a well know method for deploying large numbers of mobile sensors in unknown environments [7]. We compared our proposed EANet algorithm with ATRI over the training scenarios in Figures 2 and 3 considering different numbers of UAVs and users. We briefly describe the ATRI algorithm here, more details are in the original paper [7]. In ATRI, each UAV at the center of a circle of radius U_{ar} divides this circular region into six sectors. Recall that U_{ar} is a UAV's air-to-air communication range. ATRI then finds the nearest neighbor in each sector, if any, and computes the sum of difference vectors. If the magnitude of this sum vector exceeds a threshold value, the UAV moves in the direction of this vector, otherwise, it moves away. If users require more bandwidth than servable by the current deployment, ATRI reduces this threshold upto U_{gr}. Each UAV updates its position periodically, that is for N UAVs to be deployed, a UAV will update its position and wait till remaining $N-1$ UAVs update their positions. The next section shows and explain the experimental results obtained using EANet and ATRI.

TABLE I: Comparing percentage fitness difference between EANet and ATRI on training scenarios across UAVs and users

	156 UAVs					117 UAVs					78 UAVs				
Users	**S1**	**S2**	**S3**	**S4**	**Avg**	**S1**	**S2**	**S3**	**S4**	**Avg**	**S1**	**S2**	**S3**	**S4**	**Avg**
200	4.63	21.66	12.68	12.1	12.83	1.19	25.59	15.81	11.21	13.54	7.66	24.66	-11.48	15.75	9.30
150	2.95	16.34	11.93	9.39	10.16	6.94	21.53	16.65	18.82	15.98	8.05	33.85	-10.62	16.15	11.95
100	12.05	7.86	7.36	7.11	8.57	8.1	9.93	9.93	11.51	9.86	7.15	24.63	28.79	17.44	19.42
50	2.7	3.66	6.46	8.57	5.40	8.10	10.59	12.21	9.33	10.01	13.2	48.7	28.34	15.48	25.89



(a) (b)

Fig. 6: Comparing average bandwidth coverage (a) and number of active UAVs (b) for EANet and ATRI on four training scenarios with 156 UAVs and different numbers of users.

IV. Results and Discussion

We compare our algorithm, EANet, against ATRI and summarize results in this section. The experiments in our paper assumed that deployed UAVs use the 2 MHz communications channel with a spectral efficiency of 2.5 bps/Hz [17]. Each UAV serves a data rate of 5Mbps to users. We assume that initially, UAVs batteries are fully charged and have 1000×10^3 Joules of energy and hover at a constant altitude of 100 meters. Table III summarizes these and other experimental parameters. Users bandwidth requirements can range from simple text message communication to HD video streaming services during, for example, a health emergency. Thus, we assume that a user's maximum bandwidth requirement is 3Mpbs.

A. *Experiments on Training Scenarios*

EANet tunes potential field parameters for the 4 different training scenarios. We considered these scenarios with $156, 117$ and 78 UAVs available, and with $200, 150, 100$, and 50 users. Because of the very long fitness evaluation times, we used a small population (20) run for only 20 generations. Later on to see the effect of large population size and number of generations on fitness, we explored other population sizes upto 60, running for upto 90 generations without significant improvements in results. Thus we kept to a population size of 20 run for 20 generations as this was fast and gave good results. We use elitist selection where offspring double the parent population to 40. The best half (20) move on to subsequent generations. Single point crossover and point mutation produce offspring with probabilities of 0.95 and 0.05 respectively. The EA ran ten times with different random seeds.

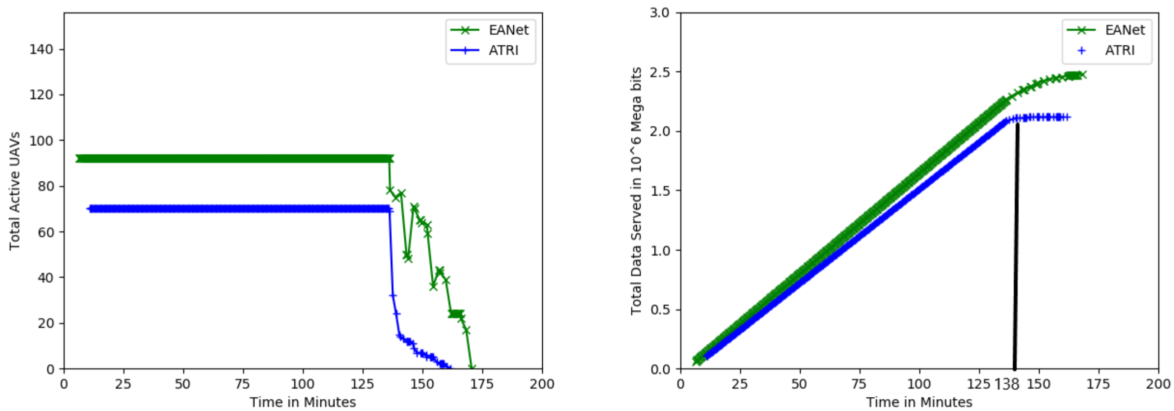
The evaluation simulation ran for 3000 time steps; 1500 time steps for the user discovery phase, phase 1, and the remaining 1500 time step for UAVs to move towards users, phase 2. The best solution from the last generation is selected for comparing with ATRI. Table I compares results for the combination of 156, 117, and 78 UAVs and 200, 150, 100, and 50 user over the 4 different training scenarios, as shown in Figure 2. We thus compare a total of $3 \times 4 \times 4 = 48$ different combination of UAVs, users, and scenarios. Each entry of Table I is computed using equation 4.

$$diff = \frac{Fitness_{EANet} - Fitness_{ATRI}}{2} * 100 \qquad (4)$$

The table summarizes our comparison results and shows the percentage difference in performance between EANet and ATRI. Positive values indicate how much better (in percentage terms) EANet performs compared to ATRI. Negative values show how much better ATRI performs. As clearly shown in Table I, EANet outperforms ATRI on 46 of the 48 possible combination of UAVs, users, and scenarios whereas ATRI perform better only in 2 such combinations. That is, EANet outperforms ATRI 96% of the time. When looking at averages

TABLE II: Comparing percentage fitness difference between EANet and ATRI on testing scenarios across UAVs and users.

	156 UAVs				117 UAVs				78 UAVs			
Users	**S5**	**S6**	**S7**	**Avg**	**S5**	**S6**	**S7**	**Avg**	**S5**	**S6**	**S7**	**Avg**
200	7.4	11.92	10.75	10.02	5.29	17.6	8.24	10.32	12.92	19.68	21.19	17.92
150	6.77	17.85	8	10.84	7.35	19.95	9.49	12.20	17.47	22.4	19.09	19.64
100	5.83	12.09	8.29	8.65	8.61	25.53	9.34	14.28	10.64	24.72	16.21	16.98
50	2.51	7.65	6.46	5.53	9.79	11.04	5.78	8.82	14.79	15.28	19.55	16.57



(a) (b)

Fig. 7: Comparing the number of active UAVs (a) and data served (b) over network lifetime for EANet and ATRI on S3 with 156 UAVs and 200 users.

TABLE III: Simulation and EA parameters.

	Parameters	Symbol	Value
UAV	Area of Interest	AOI	2000m*2000m
	Height of UAVs	U_h	100m
	Coverage Radius of UAVs	U_{gr}	100m
	A-2-A communication Range	U_{ar}	300m
	Max Speed of UAV	U_s	15 m/s
	Initial energy of UAV	E_t	1000×10^3 J
	Hovering Energy	E_h	98 J/s
	Serving per users Energy	E_u	5 J/s
EA	Max population size	$MaxPop$	20
	Max generation	$MaxGen$	20
	Crossover rate	P_x	0.95
	Mutation rate	P_m	0.05

over the four scenarios, we can see that on average we obtain between 5.4% and 25.89% performance improvement.

As stated earlier, both percentage bandwidth coverage and percentage active UAVs are combined together to get fitness. Hence, to see the effect of both on fitness, we plotted the graphs shown in Figure 6 for 156 UAVs and different number of users. The figure 6(a) shows that there is not much difference in bandwidth coverage with differing numbers of users between EANet and ATRI. The performance difference comes from the number of active UAVs where EANet uses more UAVs to serve bandwidth needs, as shown in figure 6(b). If more UAVs are used to serve a certain bandwidth, coverage lifetime should increase. Here is the link of videos for four training scenarios. The ATRI approach work does not address movement, so we wrote and used our own motion model to move the UAVs in the videos. In contrast, the EANet videos use our potential fields movement model which is a distinguishing feature of using potential fields.

B. Experiments on Testing Scenarios

Table II shows the same performance difference values on the three testing scenarios. Here, EANet outperformed ATRI in *all*, $3 \times 3 \times 4 = 36$ cases. When looking at averages over the three scenarios, we can see that on average we obtain between 5.53% and 19.64% performance improvement. This provides evidence that our results generalize and specifically that parameters tuned in training scenarios are robust to difference distributions of users in never before seen scenarios. Once again we observe that the performance difference can be attributed to longer network lifetimes as UAVs remain active longer. Here is a link to videos for three testing scenarios showing simulations for both deployment phases.

Figure 7(a) plots the number of active UAVs versus time while Figure 7(b) shows cumulative data served over the same time on the 3^{rd} training scenario as shown in Figure 2. During our experiments, the total data served by all UAVs is computed every 20 seconds and Figure 7(a) shows how total number of active UAVs decreases over a time period for both algorithms. Using ATRI, active UAVs number reduces drastically and after 138 minutes of deployment, the total active UAVs are less than 10. At 165 minutes, the total active UAVs is 0 for ATRI and at 170 minutes after deployment the total active UAVs is 0 for EANet algorithm. However, due less number of active UAVs in case of ATRI deployed network, the data served to users after 138 minutes is almost stagnated. Figure 7(b) shows the clear difference in the total data served. Larger number of active UAVs allows our proposed algorithm to deliver more data to users. The figures indicate that EANet serves more data for a longer amount of time compared to ATRI.

V. Conclusions and Future Work

In this work, we proposed an evolutionary algorithm, EANet, for adaptive UAV mesh network deployment in unknown environments. We start by distributing UAVs for maximal area coverage while maintaining the mesh network connected to the command center in order to find all users. We assume that UAVs are equipped with sensors to find users on the ground in the AOI. In the next phase, once users and their requirements are know, a set of potential fields tuned by our evolutionary algorithm guides UAVs towards users to provide bandwidth coverage for maximal time. Networks last longer when more UAVs share users' bandwidth requirements.

Results reported in this paper show that EANet on average outperforms ATRI across 48 different scenarios with varying numbers of users, UAVs, and user distributions. Moreover, although the parameters were tuned on four training scenarios used during fitness evaluation, EANet outperformed ATRI on distributions of users not seen during evolution. This indicates the potential robustness of our approach. We also note that using potential fields yields collisions free movement which has not been addressed by prior work in this area. Simulating UAVs using real-UAV speeds, acceleration, and turning radii will provide much better and usable estimates of network lifetimes. We plan to begin this work next.

In the future, we plan to deploy UAVs in the real world and consider more realistic situations where localization is a significant problem and the exact positions of UAVs during deployment will not be available to compute potential fields. We believe that our approach subsumes ATRI, as well as earlier more static approaches based on the circle point theorem and delauney triangulation [6], [7] and are working on both theoretical and experimental verification of this conjecture. For example, we are working on a theoretical derivation of potential field parameters that will result in mimicking ATRI. Next, we plan to formulate UAV network deployment as a multi-objective optimization problem and use evolutionary multi-objective optimization techniques to provide a pareto front of tradeoff choices between lifetime and bandwidth coverage. Finally, prior work has mainly addressed static users that do not move. We plan to begin addressing such unrealistic assumptions by incorporating dynamic users and user routing in simulation, and thus extending our approach to dynamic environments.

Acknowledgments

This work was supported by grant number N00014-17-1-2558 from the Office of Naval Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Office of Naval Research. This work was also supported in part by the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (USDOT/OST-R) under Grant No. 69A3551747126 through INSPIRE University Transportation Center (http://inspire-utc.mst.edu) at Missouri University of Science and Technology. The views, opinions, findings and conclusions reflected in this publication are solely those of the authors and do not represent the official policy or position of the USDOT/OST-R, or any State or other entity.

References

[1] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Mérouane Debbah. A tutorial on uavs for wireless networks: Applications, challenges, and open problems. *IEEE Communications Surveys & Tutorials*, 2019.

[2] Dalimir Orfanus, Edison Pignaton de Freitas, and Frank Eliassen. Self-organization as a supporting paradigm for military uav relay networks. *IEEE Communications Letters*, 20(4):804–807, 2016.

[3] Haitao Zhao, Haijun Wang, Weiyu Wu, and Jibo Wei. Deployment algorithms for uav airborne networks toward on-demand coverage. *IEEE Journal on Selected Areas in Communications*, 36(9):2015–2031, 2018.

[4] Xiao Zhang and Lingjie Duan. Fast deployment of uav networks for optimal wireless coverage. *IEEE Transactions on Mobile Computing*, 18(3):588–601, 2018.

[5] Amar Nath Patra, Paulo Alexandre Regis, and Shamik Sengupta. Distributed allocation and dynamic reassignment of channels in uav networks for wireless coverage. *Pervasive and Mobile Computing*, 54:58–70, 2019.

[6] Zsolt Gáspár and Tibor Tarnai. Upper bound of density for packing of equal circles in special domains in the plane. *Periodica Polytechnica Civil Engineering*, 44(1):13–32, 2000.

[7] Ming Ma and Yuanyuan Yang. Adaptive triangular deployment algorithm for unattended mobile sensor networks. *IEEE Transactions on Computers*, 56(7):946–847, 2007.

[8] Jiangbin Lyu, Yong Zeng, Rui Zhang, and Teng Joon Lim. Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607, 2016.

[9] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, and Mérouane Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8):1647–1650, 2016.

[10] Alexander C Woods and Hung M La. A novel potential field controller for use on aerial robots. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 49(4):665–676, 2017.

[11] S. Liu, S. Louis, and C. Ballinger. Evolving effective micro behaviors in real-time strategy games. *IEEE Transactions on Computational Intelligence and AI in Games*, PP(99):1–1, 2016.

[12] Sushil J Louis and Siming Liu. Multi-objective evolution for 3d rts micro. In *2018 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2018.

[13] Siming Liu, Sushil J Louis, and Christopher Ballinger. Evolving effective micro behaviors in rts game. In *2014 IEEE Conference on Computational Intelligence and Games*, pages 1–8. IEEE, 2014.

[14] Navin K Adhikari, Sushil J Louis, and Siming Liu. Multi-objective cooperative co-evolution of micro for rts games. In *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019.

[15] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. In *Autonomous robot vehicles*, pages 396–404. Springer, 1986.

[16] Rahul Dubey, Joseph Ghantous, Sushil Louis, and Siming Liu. Evolutionary multi-objective optimization of real-time strategy micro. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2018.

[17] Raj Jain, Fred Templin, and Kwong-Sang Yin. Analysis of l-band digital aeronautical communication systems: L-dacs1 and l-dacs2. In *2011 Aerospace Conference*, pages 1–10. IEEE, 2011.