Genetic Optimization Of UAV-Based Ad-Hoc Wireless Network Deployment

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

***Abstract*—Autonomous Unmanned Aerial Vehicles (UAVs) acting as wireless access points can be deployed to quickly create on-demand wireless mesh networks in remote areas or when existing communication networks fail. This paper attacks UAV-based network deployment by using a genetic algorithm to optimize the movement of UAVs deployed to create such wireless networks. Specifically, we use a set of potential fields to guide UAV deployment and genetic algorithms to optimize these potential field parameters in order to maximize bandwidth coverage and longevity of deployed networks for uniform and non-uniform user distributions. We make the common assumption that UAVs have limited sensing range and can communicate with h hop neighbors. Increasing the value of h leads to more information and subsequently better decision making. Experimental results with one hop communication on different combinations of users' distribution and number of available UAVs for deployment show that wireless networks deployed using our approach performed significantly better than the state-of-the-art UAV network deployment algorithm. Furthermore, enabling two hop communication between UAVs leads to significant performance improvement. These results hold across different scenarios with different numbers of UAVs, users, and user distributions.**

Keywords*—genetic algorithms, optimization, UAVs, wireless networks, potential fields.

I. Introduction

UAVs are gaining traction as a useful tool in many different domains including search and rescue [1], surveillance [2], cargo transport [3], surveying and mapping [4], disaster relief [5], and IoT [6]. We are specifically interested in the search and rescue application domain where the challenge is to establish on-demand connectivity in remote areas or when existing communication infrastructure fails to operate due to natural calamity or stress on the existing network [7]. In recent years, UAVs acting as flying access points have emerged as a viable alternative to provide flexible but time limited wireless infrastructure to connect people/users [8] or to collect data from Internet of Things (IoT) devices in a given Area of Interest (AOI) [9]. However, many challenges arise while deploying such UAV based ad-hoc networks including optimal UAV deployment, maintaining connectivity, and increasing the longevity of the deployed network [7].

Figure 1 shows an example wireless mesh network using eight UAVs to provide bandwidth coverage to twelve users. In this figure, UAVs are shown inside black rectangular boxes and users are shown by red rectangles. We assume that each UAV

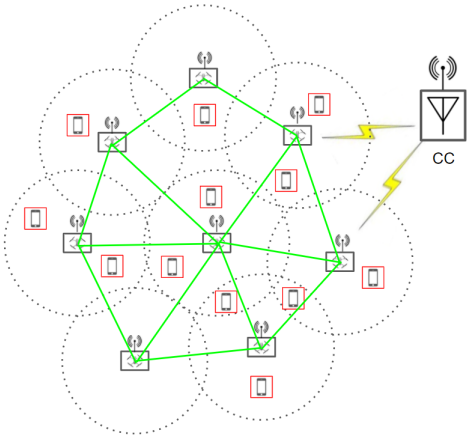


Fig. 1. An example of UAVs network deployment with 8 UAVs, 12 users, and a command center labelled "CC." UAVs are shown by black boxes and red boxes represents users.

can communicate with users within a certain range as shown by black-dashed circles and can communicate with neighboring UAVs shown by green links. A command center (CC) is placed on the right and is in direct communication with two UAVs. Each UAV can communicate with the CC either directly or through neighboring UAVs to provide bandwidth coverage to all twelve users.

Deployment of UAV based ad-hoc networks in unknown environments is a challenging problem. In unknown environments, the firs challenge is to find all users.Once all users are discovered and their positions and bandwidth requirements are known, adaptively re-deploying UAVs in real-time to serve all discovered users specifies the second challenge. To solve both problems, we divided the network deployment problem into two phases: search and service. In the search phase, UAVs are deployed to cover the entire area of interest to find all users' locations and bandwidth requirements. Given enough UAVs, several existing algorithms can position UAVs to optimally cover the area of interest in the search phase and we show how distance based potential fields may be used to mimic one of these algorithms. The second phase (service) utilizes the information gathered in the search phase to move UAVs under the influence of a set of potential fields to optimize user service. Several algorithms exist that adapt to users' requirements, but they do not perform well in diverse users'

distribution scenarios.

Optimal UAV based ad-hoc network deployment during the second phase is an NP-hard problem [10]. This paper presents a genetic optimization approach, Genetic Algorithm Network Deployment (GANet), to optimally deploy UAVs acting as wireless access points providing an ad-hoc wireless network to serve users in the area of interest. This network deployment problem has been formulated as an optimization problem with the aim of maximizing the sum of bandwidth coverage and longevity of deployed networks. In our formulation, UAV's use potential fields which are highly non-linear, to guide their movement. Genetic Algorithms (GAs) have been used extensively to evolve solutions for poorly understood non-linear problems [11] [12]. We thus use a genetic algorithm to search for and find potential field parameters that guide UAV movement in order to maximize network bandwidth coverage and longevity. Section IV of the paper explains the genetic algorithm in detail. To deal with unknown environments and users' distribution, we evolved potential field parameters on four different *training* scenarios with different user distributions (uniform and non-uniform) and measured the robustness of the best evolved potential field parameters on 100 different *test* scenarios. We also considered different levels of information exchange between UAVs to measure the differences in the performance of deployed networks. Specifically, we compared 1-hop (neighbor) UAV-to-UAV communication with 2-hop (neighbor's neighbors) communication. Experiments on *training scenarios* show that GANet performed better than adaptive triangular network deployment (ATRI) [13] under different combinations of number of UAVs and number of users on 1-hop networks. The same parameters also led to better performance than ATRI on our *testing* scenarios. Results from 2-hop UAV communications show that networks deployed with 2-hop communications performs statistically significantly better than networks with 1-hop communication. Evidence indicates that this is due better network capacity utilization, leading to longer periods that UAVs stay in the air, and thus longer network lifetimes.

The three main contributions of this paper are 1) this paper presents a new unified approach, genetic optimized potential fields, for both phases of UAV based network deployment that outperforms the state of the art under uniform and non-uniform user distributions, 2) 2-hop UAV communication statistically significantly improves problem performance and attains or gets close to optimum indicating we may not need more than 2-hop UAV communication, and 3) GANet evolved robust solutions that work well on unseen distributions of users.

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 sets up the problem, specifies our training, and testing scenarios. Next, section IV describes our algorithm, how potential fields determine UAV movement, and fitness computation. Experimental setup and results constitute section V and the last section provides conclusions.

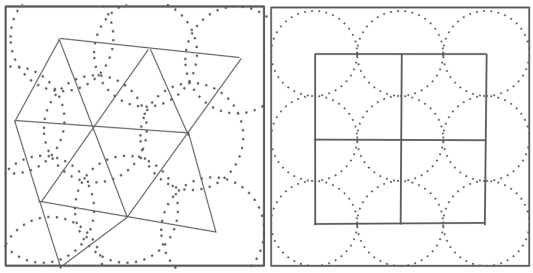


Fig. 2. Networks deployed using DT and CPT in a square AOI. Dashed circle shows the coverage of a UAV.

II. Literature Review

Many challenges arise when deploying a wireless mesh network in unknown environments. Optimal UAV placement, fast deployment, maintaining a mesh network connected to a command center, minimizing the number of deployed UAVs, maximizing the lifetime of deployed networks, routing, and channel allocation, all present significant challenges [7]. In the literature, several techniques have been introduced for positioning UAVs based on Delaunay Triangulation (DT) [13], Circle Packing Theorem (CPT) [14], and Voronoi Diagrams (VD) [15] for the search phase (also called the static network deployment phase). A network deployed using DT has no coverage gap while having minimum overlap between UAV's sensing areas. CPT computes locations of UAVs while ensuring no overlap and minimum coverage gap as shown in the figure 2. VD divides the entire AOI into different segments and places a UAV in each segment to provide bandwidth coverage to users. Such static network deployment is primarily used for area coverage. That is, we want UAVs to deploy to obtain sensor coverage of the entire area of interest and progress on this problem has several applications. Lam [16] deployed a heterogeneous sensor network using circle packing by filling the given AOI with circles of different radii corresponding to different UAV types. Lyu [17] proposed a placement optimization technique that first deploys a network using CPT and then adjusts the altitude of deployed UAVs to cover the entire AOI with fewer UAVs while providing wireless coverage to ground terminals. These are computational geometry model based techniques to compute the position of UAVs and they do not adapt to users' locations or bandwidth requirements. Our proposed GANet on the other hand adapts to both while maximizing network longevity.

For the second phase, researchers have presented modifications to make static networks more adaptive to user's requirements. Ming [13] introduced the state of the art Adaptive Triangular Network Deployment (ATRI) algorithm by taking inspiration from Delaunay Triangulation. Using one-hop neighbor's state information, ATRI generates a mesh network by adjusting the distance between neighboring UAVs in order to better share bandwidth within a given AOI. Unlike our approach, ATRI, however, does not work well when users are distributed non-uniformly or clustered in groups. In [18],

the authors initially deploy UAVs using CPT and presented a mathematical model to adjust the altitude of deployed UAVs for near optimal coverage. Increasing the altitude of UAVs increases the coverage range but decreases signal strength resulting in poorer coverage quality. In our work, UAVs are deployed at a fixed altitude and only move in two dimensional space. Bartolini [19] proposed a voronoi polygon based adaptive network deployment algorithm to deploy heterogeneous mobile sensors over a field of interest. The algorithm computes different polygons for different sensors and moves a sensor only when the sensor does not detect any users on the ground. Amar [20] presented a dynamic algorithm to serve a subregion within the AOI that requires more bandwidth. Both these papers assume uniform users' distribution while we look at non-uniform users' distribution as well. We use potential fields to deploy UAVs in both phases. Section IV of this paper describes how we use potential fields to deploy networks in the first phase to mimic DT and then re-deploy UAVs using a set of potential fields optimized by genetic algorithm to maximize bandwidth coverage and longevity during the second phase.

Potential field based real-time control of mobile agents was first introduced by Khatib in 1986 [21]. Owing to its simplicity, many researchers used potential fields to control UAVs and other autonomous agents in different domain specific tasks [22]–[24]. Howard [25] used potential fields to deploy mobile sensor networks to cover an area. In this approach, an agent experienced repelling potential fields based on distance from other agents and obstacles and moved towards unexplored areas. Poduri [26] introduced a mobile network deployment algorithm with the constraint that each agent has at least K neighbors where K is a user defined number. Poduri's paper provides evidence that potential fields can be used to generate a static mesh network similar to that generated by Delaunay Triangulation or the Circle Packing Theorem. Zhao [27] presented a centralized algorithm and a potential field based distributed algorithm for UAV deployment while maintaining connectivity among UAVs. However, the authors 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. All these approaches work well in relatively uniform distribution of users but not as well when users are distributed non-uniformly.

Closer to our work, researchers have used genetic algorithms to deploy static networks by optimally placing UAVs in the AOI. Reina [28] presented a multi-layout multi-subpopulation genetic algorithm to deploy UAVs optimally to maximize a linear combination of coverage, fault-tolerance, and redundancy. Dina [29] introduced a variable length genetic algorithm to maximize area coverage and minimize deployment cost using non-homogeneous sensors. These papers provide UAV positions as their output. In contrast, we use a genetic algorithm to evolve potential field parameters to guide UAVs. This not only gives us UAV positions, but also guides collision free UAV movement to these positions. In this paper, we present a new representation and use a genetic algorithm to optimize UAV based wireless network deployment. Our work is different from prior work in that not only do we provide for positioning information we also provide collision free movement and use the same representation, potential fields, for both search and serve phases. Before explaining how potential fields control the movement of UAVs, we first formulate the problem in the next section.

III. Problem Formulation

We formulate network deployment problem as an optimization problem and describe how each candidate solution is evaluated. Assume that a set of N homogeneous UAVs $U = \{u_1, u_2, ..., u_n\}$ need to be deployed in a given AOI of $2000 \times 2000\ meters^2$ with the Command Center (CC) located in the middle at $(1000, 1000)$. All UAVs start at the center within a $10 \times 10\ meter^2$ area and fly at a constant altitude of 100 meters. Each UAV is equipped with sensors to find users on the ground within a ground sensing range (U_{gr}) of 100 meters and can communicate with its neighbors within an air to air communication range (U_{ar}) of 300 meters. A total of m users each with a position and a bandwidth requirement (p_i, b_i) are distributed over the AOI. Thus users can be denoted by the list of pairs: $\{(p_1, b_1), (p_2, b_2), ..., (p_m, b_m)\}$. These simulation parameters can be easily changed. This paper treats the network deployment problem as a search problem and thus, channel modelling for communication between UAVs and UAV-user are not discussed.

Since users' locations and their bandwidth requirements are not known prior to finding them, UAVs first need to search and find all users and their bandwidth requirements by covering the AOI and then second, move to serve users that have been found and avoid areas devoid of users. Thus, the network deployment problem has been divided into two phases; search and service. The service phase is an optimization problem to maximize bandwidth coverage and longevity of deployed networks. To maximize bandwidth coverage, UAVs must be placed such that all users are covered, and to maximize longevity the energy consumption of UAVs must be reduced.

Each UAV has a limited initial energy of 10^6 joules and consumes energy while hovering/moving and providing data to users. Intuitively, if more UAVs share the bandwidth demanded by users, the per UAV bandwidth service can be reduced leading to less energy consumption and longer UAVs flight times. Thus, the objective is to deploy UAVs to maximize bandwidth coverage while sharing bandwidth demands. We classify each UAV either as an Active UAV (AU) or an Inactive UAV (IU) based on whether or not the UAV is serving a user. More specifically, if a UAV finds users within its ground sensing range, U_{gr}, the UAV is an active UAV, otherwise an inactive UAV. The number of active UAVs is proportional to network lifetime. The more AUs, the more bandwidth can be distributed among AUs leading to less power consumption per AU and thus longer battery life to stay aloft and keep the network alive. A variable $\alpha_i \in \{0, 1\}$ represents the status of UAVs where α_i is 1 for active and 0 for inactive UAVs. Equation 1 specifies the objective function where i sums over N UAVs and u_{bi} represents the bandwidth served by the i^{th} UAV, b_j is the bandwidth requirement of the j^{th} user, m is

the number of users, and N is the total number of UAVs.

$$Maximize \quad f = \left[\frac{\sum_{i=1}^{N} u_{bi}}{\sum_{j=1}^{m} b_j} + \frac{\sum_{i=1}^{N} \alpha_i}{N} \right] \tag{1}$$

In equation 1, the first term counts the fraction of bandwidth served over the total demand and the second term counts the fraction of active UAVs. Each term in the equation can have a maximum value of 1 and thus the maximum objective function value (their sum) will be 2. When computing the objective function using equation 1, several constraints must be satisfied. Each UAV must be positioned within the specified AOI, thus the UAV's x and z coordinates must be $\in (0, 2000)$[1]. A UAV can provide a maximum of u_{bmax} to users, and thus $u_{bi} \leq u_{bmax}$. UAVs can either communicate with their direct neighbors (1-hop) or with neighbors of neighbors (2-hop), and thus a UAV can communicate with other UAVs up to a distance of $2 \times U_{ar}$. UAVs use the 2 MHz communications channel with spectral efficiency of 2.5 bps/Hz [30] and a UAV can serve a data rate of 5 Megabits per second (Mbps) to users. Users' bandwidth requirements can range from simple text message communication to HD video streaming services during, for example, a health emergency. We thus assume that a user's maximum bandwidth requirement is 3 Mpbs which is enough for video communication.

IV. Methodology

We start by explaining our elitist genetic algorithms, and then describe different scenarios (training and testing) to evolve robust solutions. This section also provides the details of UAV movement, and how potential fields are used for each of the two phases of network deployment in our simulation.

A. Genetic Algorithms

Genetic algorithms have been used extensively to evolve solutions for poorly understood non-linear problems [11] [12]. Our approach works in two phases and uses potential fields to control the movement of UAVs in both phases. Since potential fields are highly non-linear, tuning the potential field parameters is difficult, and thus we used a $(\mu + \lambda)$ elitist genetic algorithm [31] to optimize parameters. Note that we use a GA only in the second phase of network deployment as the first phase of network deployment is a relatively simpler problem. The genetic algorithm has many parameters that determine the GA's time to convergence and solution quality. Much empirical work has shown that a good balance between randomized exploration of the search space and time to convergence is required. We thus performed a number of prototyping experiments to explore different population sizes, number of iterations (or generations), types of selection strategies, crossover types, and mutation algorithms and associated probabilities. These preliminary experiments led us to choose an elitist selection strategy with binary tournament selection, real crossover, and polynomial mutation. Our elitist genetic algorithm shown in Algorithm 1 searches through the space of potential field parameter values in order to maximize the fitness computed using equation 1.

[1]Our simulation uses the xz plane for the horizontal plane.

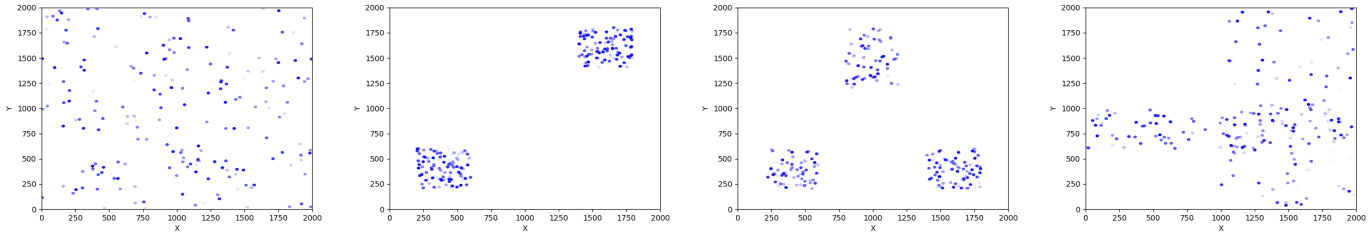
Algorithm 1: An Elitist Genetic Algorithm

Input : $Pop, Gen, P_x, P_m, \lambda, H$
Output: *Best Solution*

$P_0 \leftarrow Initialize(Pop)$
$Evaluate(P_0, H)$
P_c = []
for *t in Gen* **do**
  for *i in* λ **do**
    $p_1, p_2 \leftarrow SelectParents(P_t)$
    $c_1, c_2 \leftarrow Crossover(p_1, p_2, P_x)$
    $c_1 \leftarrow Mutate(c_1, P_m)$
    $c_2 \leftarrow Mutate(c_2, P_m)$
    $P_c.add(c_1, c_2)$
    $i = i + 2$
  end
  $Evaluate(P_c, H)$
  $P_{t+1} \leftarrow NextGenIndividuals(P_t, P_c)$
  P_c = []
end

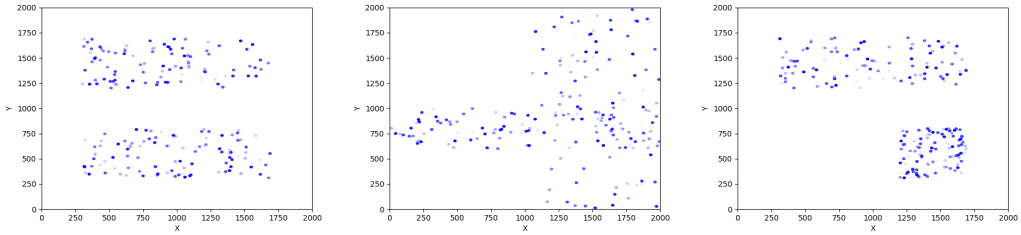
In Algorithm 1 Pop is the population size, Gen is the maximum number of iteration for evolution, P_x, P_m are probabilities of crossover and mutation, and H is the number of hops for communication between UAVs. We use $(\mu + \lambda)$ elitism where μ is the population size (Pop) and λ is number of children generated through recombination and we set $\lambda = \mu$. Initially, the GA randomly generates a parent population (P_0) of candidate solutions (CS) and evaluates the fitness of each CS using our UAV-based network deployment simulator. To generate the next population, two candidate solutions (p_1, p_2) are selected from the parent population using tournament selection with tournament size of two [24]. Next, these two solutions (p_1, p_2) exchange information with each other through a crossover operator with a probability of P_x and produce two children c_1, c_2. Using the mutation operator, each gene of c_1, c_2 is mutated with a small probability of P_m. This process of producing children continues until λ children are produced. Each child solution is then evaluated and assigned a fitness using equation 1. In $(\mu + \lambda)$ elitism, the μ best candidate solutions are selected from the combined parent and offspring pool of $(\mu + \lambda)$ individuals in the population. In Algorithm 1, `NextGenIndividuals` does this down selection operation and the process of evolution goes on for Gen number of iterations.

We aim to evolve high quality and *robust* solutions for wireless network deployment. However, earlier work has shown that solutions evolved on one scenario with a particular set of parameters may not be robust [23], [32], [33]. That is, in our case, potential field parameters evolved on one scenario may over-fit and thus not work well in other scenarios. We therefore created four *training* scenarios with different user distributions as shown in figure 3. In these scenarios 200 users



(a) T_{r1} (b) T_{r2} (c) T_{r3} (d) T_{r4}

Fig. 3. The four training scenarios with different users' distribution. Blue colored dots show the locations of users, darker the dots more bandwidth required and vice-versa.



(a) T_{e1} (b) T_{e2} (c) T_{e3}

Fig. 4. The three testing scenarios distinct from training scenarios and never seen during parameter evolution.

are distributed uniformly (T_{r1}), in clusters (T_{r2}, T_{r3}), and in the right and center region of a 2000×2000 $meter^2$ AOI. With a UAV ground sensing range of 100 meters, we need 156 UAVs to cover the entire AOI without coverage gaps in order to find all users. The genetic algorithm seeks to maximize the average objective function value, computed using equation 1, over all four scenarios. Once the genetic algorithm evolves good parameter values, we test the robustness of these potential field parameters on 100 different unseen *test* scenarios. Three of these test scenarios are shown in figure 4 in order to see that the distribution of users in these scenarios is different from the distribution in training scenarios. The aim is to show that the genetic algorithm evolved potential field parameters work across a wide range of user distributions in the AOI and our results (Section V) show that using average performance over the four training scenarios translates robustly to testing scenarios.

B. UAV Movement Modeling

Initially, each UAV starts near the center of the AOI and has zero speed. The speed (s) increases, with a constant acceleration (r_s) of 0.1 $meter/sec^2$ until speed becomes the max speed of 15 $meter/sec$, using equation 2. Here Δt is the simulation time step interval.

$$s = s \pm r_s \Delta t \tag{2}$$

Each UAV moves under the influence of a set of potential fields and the direction of the vector sum of these potential fields provides the desired heading or direction for the UAV ($\in (0, 2\pi)$) to move along. A UAV (k^{th}) experiences attractive ($\vec{P_a}$) and repulsive ($\vec{P_r}$) potential fields from neighboring UAVs within the range of U_{ar} and given by equation 3. Each potential field is of the form cd^e where d is distance and has two optimizable parameters (c, e) that determine field effect.

$$\begin{aligned} \vec{P_k} &= \sum_{i=0}^{n} \vec{P_a} + \sum_{i=0}^{n} \vec{P_r} \\ \vec{P_k} &= \sum_{i=0}^{n} \vec{a_{ki}} c_a d_{ki}^{e_a} + \sum_{i=0}^{n} \vec{a_{ik}} c_r d_{ki}^{e_r} \end{aligned} \tag{3}$$

Here $\vec{a_{ki}}$ is a unit vector pointing from the k^{th} UAV to the i^{th} neighbor and $\vec{a_{ik}}$ is the unit vector pointing from i^{th} neighboring UAV back to the k^{th} UAV. n is the number of h-hop neighbor UAVs, $d_{ki} = d_{ik}$ is the distance between the k^{th} and i^{th} UAVs. c_a, c_r are potential field coefficients and e_a, e_r are potential field exponents. The GA optimizes these coefficients and exponents to specify UAV movement and $\vec{P_k}$'s direction specifies the k^{th} UAV's heading. In our model, UAVs change their current heading to this new desired heading instantaneously. Using speed, heading, and equation 4 we compute a UAV's velocity ($\overrightarrow{vel}$) and position ($\overrightarrow{pos}$).

$$\begin{aligned} \overrightarrow{vel} &= (s \times cos(heading), 100, s \times sin(heading)) \\ \overrightarrow{pos} &= \overrightarrow{pos} + \overrightarrow{vel} \times \Delta t \end{aligned} \tag{4}$$

Having specified UAVs movement model, next we describe how to use potential fields for area coverage during the first phase.

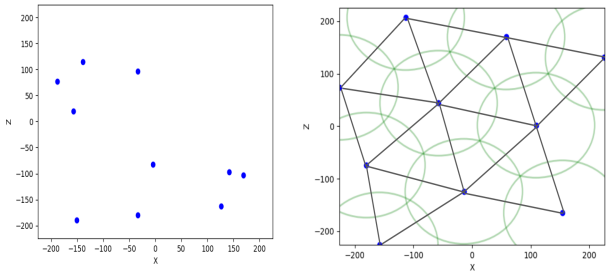


Fig. 5. 10 randomly distributed UAVs (a), and each UAV moves under the influence of attractive and repulsive potential fields with parameters ($c_a = 1, e_1 = 1, c_r = 3 \times \sqrt{3} \times 100^3, e_r = -2$). Figure (b) shows that using potential fields, we can get network deployment similar to networks inspired from DT.

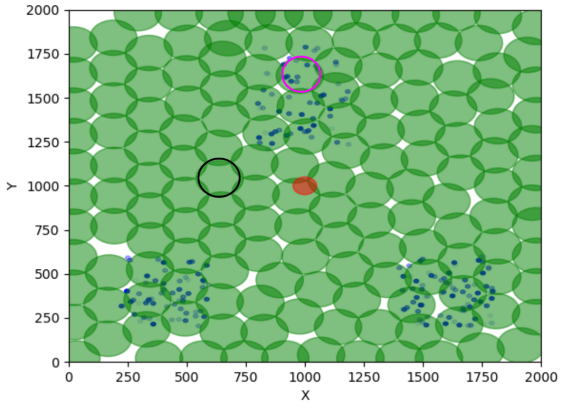


Fig. 6. Deployment of UAVs at the end of the first phase on the third training scenario. Blue dots represent users, green circles represent UAVs' coverage areas, and the red circle represents the location of the command center. A black circle shows an inactive UAV and the magenta circle shows an inactive UAV.

C. Network Deployment

1) First Phase: Search: In this phase, UAVs move to cover the entire AOI to find users' distribution. This is also called static network deployment and we use potential fields to deploy UAVs to maximize area coverage with minimal overlap. Since Delaunay Triangulation maximizes area coverage with minimal overlap, the aim is to mimic DT style network deployment with potential fields. We thus use Ming's [13] Delaunay Triangulation based result that shows that the distance between any two adjacent UAV must be $\sqrt{3}\, U_{gr}$ to cover the AOI with minimum coverage overlap. We thus need to find values of c_a, c_r, e_a, e_r such that when $d_{ki} = \sqrt{3}\ U_{gr}$ the magnitude of the attractive potential field equals the magnitude of the repulsive potential field ($\|\vec{P}_a\| = \|\vec{P}_r\|$). This is shown by equation 5.

$$\begin{aligned} \hat{a_{ki}} c_a d_{ik}^{e_a} &= \hat{a_{ki}} c_r d_{ik}^{e_r} \\ c_a(\sqrt{3}U_{gr})^1 &= c_r(\sqrt{3}U_{gr})^{-2} \\ c_r &= c_a(\sqrt{3}U_{gr})^3 \end{aligned} \tag{5}$$

In the equation, the directions of unit vectors $\vec{a_{ki}}$ and $\vec{a_{ik}}$ are opposite to each other ($\vec{a_{ik}}$ = - $\vec{a_{ki}}$). In addition, earlier work in potential fields based movement has shown that the attractive potential should be proportional to distance and the repulsive potential field should be inversely proportional to distance squared to provide collision free cohesive movement [34]–[37]. Thus we set $e_a = 1$, $e_r = -2$, and $d_{ik} = \sqrt{3}U_{gr}$ and ignore the unit vectors in equation 5 to find the relationship between c_a and c_r. In simulation experiment, $U_{gr} = 100$ meters and, for simplicity, assumed $c_a = 1$ to derive the value of c_r to be $3\sqrt{3}\ 100^3$ by simplifying equation 5. These parameter values of $c_a = 1, e_1 = 1, c_r = 3\ \sqrt{3}\ 100^3$, and $e_r = -2$ were then used to control the movement of 10 randomly distributed UAVs as shown in figure 5(a) and the resulting deployment covering the AOI is shown in figure 5(b), similar to figure 2(a).

Noting the similarity to figure 2, figure 5 shows that with these parameter values, UAVs moving under the influence of potential fields can be used to mimic DT and cover the entire area with minimal overlap. Also, there exist multiple sets of values of the coefficients and exponents that will result in similar coverage. Assuming we can, in general, directly derive potential field parameter values for the search phase we focus more on the service phase in the rest of the paper. Using the same potential field parameter values, we moved to our main scenarios and deployed 156 UAVs in an AOI of 2000×2000 $meter^2$ where UAVs start within an area of 10×10 $meter^2$ near the center of the AOI. At the end of the first phase, UAVs are distributed over the given AOI as shown in figure 6. In the figure, users, represented with blue dots, are distributed in three clusters. Green circles show UAV coverage areas, the red circle represents the command center. A black circle shows a UAV serving no users within its ground sensing range (U_{gr}) and is an example of an inactive UAV. Similarly, the magenta circle shows a UAV serving multiple users and is an example of an active UAV. Increasing the number of UAVs that serve the same users leads to bandwidth sharing that decreases per active UAV energy consumption and thus increases network longevity. The second phase seeks to increase the number of active UAVs by moving UAVs from areas with lower bandwidth requirements (fewer users) to areas with higher bandwidth requirements (more users) while maintaining network connectivity with the command center.

2) Second Phase: Service: In this phase, a different set of potential fields optimized by a genetic algorithm re-deploys UAVs to serve found users better. Algorithm 2 specifies the Genetic Adaptive Network Deployment Algorithm (GANet) that runs the simulation to compute the fitness of candidate solutions. That is, the *Evaluate* function of Algorithm 1 runs Algorithm 2 to evaluate each candidate solution's fitness. Once computed, this fitness drives evolutionary optimization.

In Algorithm 2, UAVs and Users are lists of available UAVs and users respectively. The simulation runs for a maximum number of time steps ($MT = 1500$) on each training scenario. As mentioned earlier, we use four training scenarios and thus MaxScenarios (MS) is four. At each time-step (t), `AssociateUsers` scans users within U_{gr} for each UAV in the list and computes these users' bandwidth consumption.

Algorithm 2: Evaluation and fitness computation

Input : $UAVs, Users, CS, MS, MT, H$
Output: fit

```
fit = 0;
for scenario in MS do
    t, bwc, AUs = 0;
    while t ≤ MT do
        AssociateUsers(UAVs, Users);
        FindNeighbors(UAVs);
        ActivePotentials(H, CS);
        Dir = ComputeDir();
        MoveAll(Dir);
        t=t+1;
    end
    for i in UAV do
        bwc += u_bi;
        AUs += α_i;
    end
    fit += bwc / MaxBW + AUs / UAVs;
end
fit = fit / MS;
return(fit);
```

`FindNeighbors` finds other UAVs within U_{ar} and records them as neighbors. `ActivePotentials` uses the number of hops and potential field parameter values supplied by a candidate solution from the genetic algorithm's population, to compute potential field values. Now that the potential field information needed is available, `ComputeDir` computes potential field directions at each UAV by summing all potential fields acting on a UAV using equation 6 and stores this in `Dir`. Next, `MoveAll(Dir)` moves each UAV in the direction computed. After MT time steps, the algorithm computes the fitness for each training scenario by summing the bandwidth coverage fraction and active AU fraction. Finally, once the evaluation on all four scenarios is over, the algorithm returns the average of these fitness values as the fitness of this candidate solution to the genetic algorithm.

$$\vec{P_k} = \vec{P_b} + \sum_{i=1}^{h} (\vec{P_{u_i}} + \delta_r \vec{P_{r_i}}) + \delta_c \vec{P_c} \tag{6}$$

Note that in equation 6, h refers to the number of hops for communication between UAVs. The vector sum of the potential fields for UAV k given by $\vec{P_k}$ provides a desired heading for the k^{th} UAV to turn towards. Each term in the equation describes a potential field whose parameters need to be optimized (or tuned). Since we want UAVs to be attracted towards users based on their bandwidth requirements, we use an attractive potential field, ($\vec{P_b}$), to model this attraction. To reduce per UAV bandwidth coverage and save energy, the k^{th} UAV uses an attractive potential field towards neighboring UAVs whose magnitude depends on the neighbors bandwidth load ($\vec{P_u}$). At the same time, to avoid collisions, a repulsive potential field, $\vec{P_r}$, ensures that UAVs do not collide with each other. This potential field only applies to UAVs that are less than d^* distance away and we use the Dirac delta function, δ_r to model this. $\delta_r (\in \{0, 1\})$ has value 1 when UAV j is less than d^* distance away and 0 otherwise. When a UAV has no users within its user sensing range and no UAVs within U_{ar}, that is, the UAV is inactive, we want it to move towards the command center as a default behavior. We model this with an attractive potential field, $\vec{P_c}$, based on distance to the command center. δ_c is another Dirac delta function to ensure that this potential field only acts on inactive UAVs. Assuming UAVs communicate only with one-hop neighbors, equation 6 can be rewritten as equation 7 given below.

$$\vec{P_k} = \sum_{i=0}^{m_k} c_1 \vec{b_i}^{e_1} + \sum_{j=0}^{n_k} (c_2 \vec{u_{kj}}^{e_2} + c_3 \vec{r_{kj}}^{e_3}) + c_4 \vec{s_k}^{e_4} \tag{7}$$

Here c_1, c_2, c_3, c_4 are the coefficients and e_1, e_2, e_3, e_4 the exponents of the potential fields in equation 7. The first summation is over m_k which is the number of users within k^{th} UAV sensing range while the second summation is over n_k which is the number of one hop neighbors. $\vec{b}$ is the bandwidth required by the i^{th} user and points towards the user. $\vec{u}$ is the bandwidth being served by the j^{th} 1-hop neighbor of k^{th} UAV and points towards this neighbor. $\vec{r}$ is the vector difference between the positions of UAV k and UAV j and points from j to k. In the last term, $\vec{s}$ is the vector difference between the command center position and UAV k pointing from k to the command center. Including four coefficients, four exponents, and d^*, a total of nine parameters need to be optimize. When UAVs are allowed to communicate with upto two-hop neighbors, equation 6 can be re-written as equation 8.

$$\begin{aligned} \vec{P_k} \quad = \quad & \sum_{i=0}^{m_k} c_1 \vec{b_i}^{e_1} + \sum_{j=0}^{n_k} (c_2 \vec{u_{kj}}^{e_2} + c_3 \vec{r_{kj}}^{e_3}) \\ & + \sum_{j'=0}^{n_{k'}} (c_4 \vec{u_{kj'}}^{e_4} + c_5 \vec{r_{kj'}}^{e_5}) + c_6 \vec{s_k}^{e_6} \end{aligned} \tag{8}$$

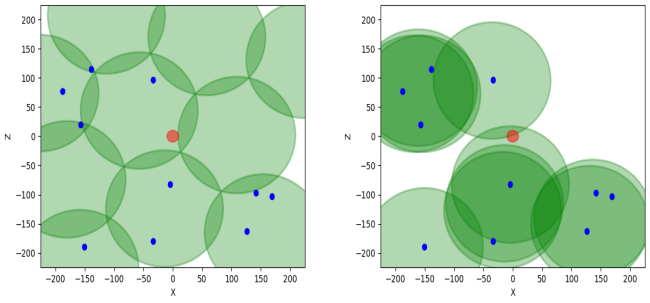
Here $n_{k'}$ is the number of two hop neighbors, that is, the number of one-hop neighbors' neighbors. Note that for $h = 1$ and $h = 2$, the k^{th} UAV experience 4 and 6 different potential fields respectively. In general for h-hop communication, the k^{th} UAV will experience $2(h + 1)$ different potential fields. Since each potential field has two tunable parameters (c, e), a total of $2 \times 2(h + 1)$ potential field parameters will need to be optimized. Due to the non-linearity of these potential fields, optimizing these parameters is difficult and we thus use a genetic algorithm. Coefficients and exponents are encoded in chromosomes, where $c's \in (-8192, 8192)$ with a precision of 1, $e's \in (-5.12, 5.12)$ with precision 0.01, and d^* ranges between $0 \leq d^* \leq 2^8$. We then compared deployed network performance using GANet against the performance of the current state-of-the-art ATRI algorithm.

V. Results and Discussion

This section starts by evaluating genetic algorithm parameter tuning performance on a simple tractable problem by comparing genetic algorithm performance against a random exhaustive search and a hill climber (gradient ascent). Once we show that

TABLE I. COMPARING THREE SEARCH ALGORITHMS

Techniques	Evaluations
Hill Climber	105
Exhaustive Search	92
Genetic Algorithm	**39**



(a) Initial positions (b) Optimal positions

Fig. 7. UAV deployment within an AOI with 10 users (blue dots). Initial positions covering the area (a), final positions focusing on user locations and bandwidth sharing (b).

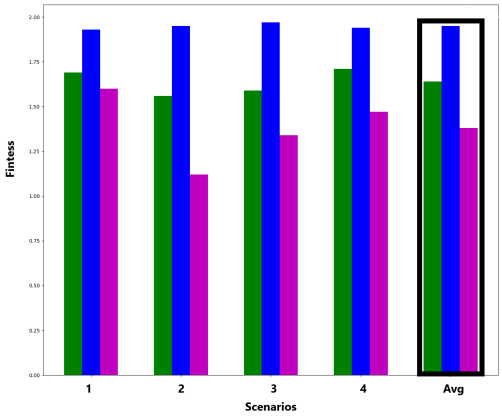


Fig. 8. Comparing GANet one-hop (green), two-hop (blue) versus ATRI (magenta) on four training scenarios. The rightmost bars show averages across the four scenarios.

the genetic algorithm finds the optimum significantly more quickly than either of the other algorithms, we run the GA on the larger, more realistic problem as in the four training scenarios.

A. Experiments on A Simple Test Problem

We begin with a test problem to deploy 10 UAVs in a small AOI of 450×450 $meter^2$ where the command center is located at $(0, 0)$. We want to provide bandwidth coverage to 10 users as shown in Figure 7(a). Using one-hop information, each UAV moves in a direction computed using equation 7. On this simple problem, we applied three search techniques; an exhaustive search, a hill climber based search technique, and our genetic algorithm, for finding optimal potential field parameters that result in the optimum fitness of 2. Our exhaustive search randomly samples points in the search space of parameter values and computes their fitness until it finds optimal fitness of 2. Exhaustive search is guaranteed to find the optimal solution, if one exists. We also ran hill climber and GA for searching optimal potential field parameters. Since the algorithms are randomized, we ran 10 times and use averages for this performance comparison. Table I compares the performance of the three algorithms and shows that the genetic algorithm finds the optimum in significantly fewer evaluations than the others. The genetic algorithm (39) is more than twice as fast as its nearest competitor (92).

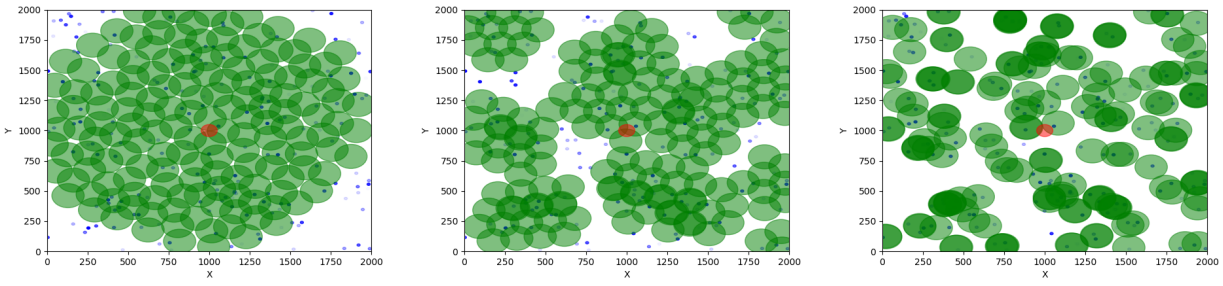
The results suggest that this simple problem is multi-modal with multiple different optimal solutions. Starting with UAV positions (center of green circles) at the end of the search phase shown in Figure 7(a) the following parameter values found by the genetic algorithm, $c_1 = 50, e_1 = 0.5, c_2 = 0.1, e_2 = 0.1, c_3 = 7000, e_3 = -5, c_4 = 0.1, e_4 = 0.1$, resulted in the optimal UAV deployment (with fitness 2) shown in figure 7(b). Note that with changes in UAV's attributes, users' requirements, or the size of AOI, the complexity and scale of the problem changes and we will need to run the genetic algorithm again. These results show the effectiveness of GA for searching potential field parameters, we move to a larger more realistic problem and compare performance against the ATRI algorithm of Ming [13].

B. Evolving Solution On Larger Problems

1) Experiments on Training Scenarios: We started with 200 users distributed as shown in figure 3, 156 UAVs, and one hop communications. We assume that UAVs start within an area of 10×10 m^2 near the command center. During the first phase each UAV moves for 1500 time steps to maximize area coverage and find users. Figure 6 shows UAV positions at the end of the first phase on the third training scenario. The figure shows that a few UAVs (who found users within their coverage range) have started providing bandwidth coverage demanded by users while many other UAVs hover idle throughout the coverage area. In the second phase of network deployment, we aim to move UAVs towards different sub-regions in the AOI that demand wireless bandwidth. Simulation runs for 1500 time steps for UAVs to move towards users under the influence of a GA optimized set of potential fields.

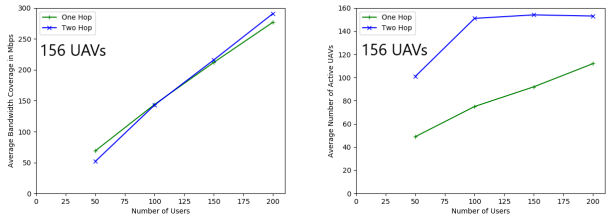
Our final genetic algorithm parameter values were taken after a grid search through genetic algorithm parameter values with populations sizes ranging from 10 to 60, the number of generations from 10 to 90, and probabilities of crossover and mutation based on values and ranges found in the literature [38]. The potential field parameters were encoded in a population of 20 individuals and GA evolved solutions on the four training scenarios for 20 generations. We encoded real valued parameters, and from the results of preliminary experiments, used simulated binary crossover with probability 0.95 and polynomial mutation with probability 0.05.

As a common practice in the GA community when dealing with long run time fitness computations, we ran the GA 10



(a) ATRI (fitness=1.60) (b) $Best_{1h}$ (fitness=1.69) (c) $Best_{2h}$ (fitness=1.93)

Fig. 9. Comparing network deployment on our first training scenario. ATRI (a) versus GANet 1-hop (b) and 2-hop (c). GANet obtains better performance by focusing more on areas with users and avoiding areas with no users.



(a) Average bandwidth coverage (b) Average number of AUs

Fig. 10. Comparing the average (a) bandwidth coverage, and (b) number of AUs of $Best_{1h}$ and $Best_{2h}$ on training scenarios.

times with different random seeds. The best solution over these 10 runs, ($Best_{1h}$, best one hop fitness) achieved a fitness of 1.64 out of 2. We observed that the network deployed using $Best_{1h}$ contains a number of inactive UAVs which contribute to the reduced fitness. Thus, to improve performance, we next tested with two hop communication. Results show that the fitness of the best solution obtained with two hop UAVs ($Best_{2h}$) was 1.95 which is significantly better than one hop and is near-optimal. For comparison, the same experiments were conducted using ATRI, and the deployed network achieved a performance/fitness of 1.38. Figure 8 shows the fitness obtained using $Best_{1h}$, $Best_{2h}$, and ATRI on the four training scenarios as well as the average performance across the four scenarios. The green and blue bars in figure 8 show $Best_{1h}$ and $Best_{2h}$ performance while magenta shows ATRI performance. The figure shows that GANet network deployment performs better than ATRI across all four training scenarios with two hop GANet providing more significant gains.

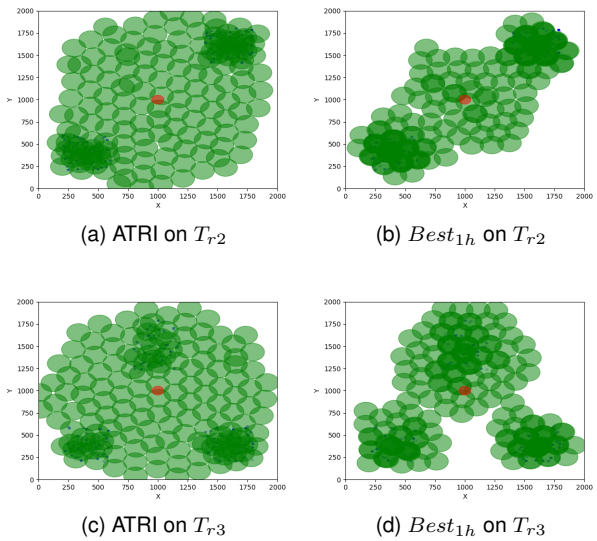
The visualizations in figure 9 help explain the differences in performance on the first training scenario. Figure 9 (a) shows a network deployed using ATRI with fitness of 1.60 and we can see that although the UAVs cover the area well, they do not adapt well to the current user distribution. Only 120 of 156 UAVs are active, and the rest (36) are inactive. These 120 active UAVs collectively provide a 245.7 Mbps data rate to users. On average, therefore, the per AU data rate is $245.7/120 \approx 2.04$ Mbps. Similarly, Figure 9 (b) shows the network deployed using $Best_{1h}$ with fitness of 1.69, has 132 active UAVs and provides 250 Mbps to users. The figure shows that GANet deployed UAVs more closely follow the user distribution and thus result in more active UAVs. On average, each active AU provides a data rate of $250/132 \approx 1.89$ Mbps. Finally, Figure 9 (c) shows the network deployed using $Best_{2h}$ with a fitness of 1.93. UAVs cluster around users and the number of active UAVs has now increased to 152 and collectively serve a data rate of 282.7 Mbps with each active UAV serving a data rate of $282.7/152 \approx 1.85$ Mbps. These figures provide evidence that GANet with one hop and two hop communication performs better than ATRI on T_{r1}.

To tease out whether bandwidth coverage or longevity is the key factor for the performance difference between 1-hop and 2-hop UAV communication, we plotted the graph of the average bandwidth coverage and the average number of active UAVs on training scenarios with 156 UAVs and different numbers of users. Figure 10(a) shows average bandwidth coverage for $Best_{1h}$ and $Best_{2h}$ for $50, 100, 150$, and 200 users on the four scenarios and we see that there is no significant difference in bandwidth coverage. Figure 10(b) shows the number of active UAVs. Here we see significantly larger numbers of active UAVs with $Best_{2h}$ compared to $Best_{1h}$ thus providing evidence that the difference in the performance must be caused by the difference in the number of active UAVs. ATRI on the other hand tries to minimize movement away from the Delaunay triangulated locations and works well with more uniform user distributions.

Figure 11 shows networks deployed using ATRI and $Best_{1h}$ on the second (T_{r2}) and third (T_{r3}) training scenarios as shown by figure 3 (b,c). Tr_2 has two user clusters and ATRI's performance is 1.12 which is 28.2% lower than $Best_{1h}$'s and 42.5% lower than $Best_{2h}$'s performance. Similar behavior can be seen for Tr_3 as well. ATRI deployed UAVs are more uniformly distributed across the AOI while GANet UAVs have learned parameter values that enable them to cluster around users while maintaining connectivity to the command center. On T_{r2} and T_{r3} per AU data rates with ATRI are 4.54 and 3.76 Mbps respectively. Whereas per AU data rate with $Best_{1h}$ are lower at 3.09 and 2.86 Mbps respectively, and with $Best_{2h}$, still lower with 1.99 and 1.88 Mbps respectively.

TABLE II. COMPARING FITNESS OF $Best_{1h}$, $Best_{2h}$, AND ATRI WITH DIFFERENT NUMBER OF UAVS AND USERS ON TRAINING SCENARIOS.

Users	Methods	Tr_1	Tr_2	Tr_3	Tr_4	Average	Tr_1	Tr_2	Tr_3	Tr_4	Average	Tr_1	Tr_2	Tr_3	Tr_4	Average
156 UAVs							117 UAVs					78 UAVs				
200	$ATRI$	1.60	1.12	1.34	1.47	1.38	1.50	1.02	1.29	1.40	1.30	1.32	0.76	1.13	1.26	1.11
200	1 hop	1.69	1.56	1.59	1.71	1.64	1.52	1.53	1.61	1.62	1.57	1.47	1.25	0.98	1.57	1.30
200	2 hop	**1.93**	**1.95**	**1.97**	**1.94**	**1.95**	**1.64**	**1.88**	**1.90**	**1.84**	**1.86**	**1.65**	**1.53**	**1.59**	**1.69**	**1.62**
150	$ATRI$	1.52	1.11	1.29	1.43	1.34	1.38	1.09	1.28	1.33	1.27	1.20	0.71	1.14	1.28	1.08
150	1 hop	1.58	1.44	1.53	1.62	1.54	1.52	1.52	1.62	1.71	1.59	1.36	1.38	0.93	1.60	1.32
150	2 hop	**1.97**	**1.96**	**1.95**	**1.93**	**1.96**	**1.69**	**1.88**	**1.80**	**1.91**	**1.82**	**1.63**	**1.66**	**1.64**	**1.66**	**1.65**
100	$ATRI$	1.34	1.13	1.24	1.42	1.28	1.26	1.40	1.28	1.34	1.25	1.10	0.81	0.93	1.23	1.02
100	1 hop	1.58	1.29	1.39	1.55	1.46	1.42	1.33	1.48	1.57	1.45	1.24	1.31	1.50	1.58	1.41
100	2 hop	**1.84**	**1.98**	**1.93**	**1.98**	**1.93**	**1.78**	**1.86**	**1.78**	**1.91**	**1.84**	**1.61**	**1.49**	**1.66**	**1.78**	**1.64**
50	$ATRI$	1.18	1.13	1.16	1.20	1.16	1.05	1.06	1.12	1.06	1.07	0.81	0.37	0.71	0.84	0.69
50	1 hop	1.23	1.20	1.29	1.38	1.27	1.22	1.27	1.36	1.25	1.27	1.08	1.34	1.27	1.15	1.20
50	2 hop	**1.32**	**1.92**	**1.29**	**1.64**	**1.54**	**1.51**	**1.77**	**1.85**	**1.60**	**1.68**	**1.21**	**1.70**	**1.67**	**1.55**	**1.53**



(a) ATRI on T_{r2} (b) $Best_{1h}$ on T_{r2}

(c) ATRI on T_{r3} (d) $Best_{1h}$ on T_{r3}

Fig. 11. Network deployments of $Best_{1h}$ and ATRI.

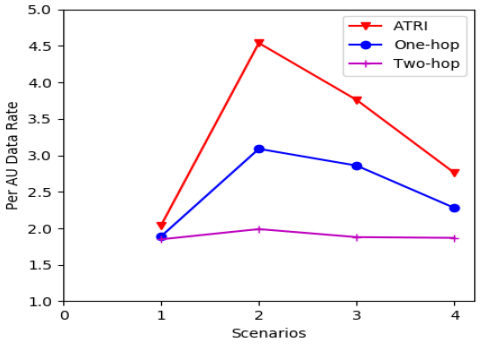


Fig. 12. Comparing per AU data rate on four training scenarios.

Lower data rates correspond to less energy consumption and longer network lifetimes. Figure 12 shows the per AU data on the training scenarios where ATRI per AU data rate is highest and GANet 2-hop is lowest. Also note that on the first training scenario where users are distributed more uniformly, ATRI's performance is comparable to GANet.

Note that $Best_{1h}$ and $Best_{2h}$ were evolved with 156 UAVs and 200 users on the four training scenarios but an earlier plot, figure 10 depicted their performance with different numbers of users. This shows that GANet finds parameter values that work on scenarios that have not been used for training and provides early evidence that evolved solutions may be robust. To further investigate robustness, we systematically varied the numbers of UAVs, the numbers of users and computed performance on the four training scenarios. Table II shows performance obtained using $Best_{1h}$, $Best_{2h}$, and ATRI with $156, 117, 78$ UAVs and $200, 150, 100, 50$ users on the four training scenarios. $Best_{2h}$ performed statistically significantly better than $Best_{1h}$ with *p-value* less than < 0.00002 while $Best_{1h}$ outperformed ATRI with $p < 0.008$. This provides evidence that even with different numbers of users and UAVs, our approach works well on scenarios with similar user distributions. In the next subsection, we provide details on experiments on *testing* scenarios that vary the distributions of users to better investigate robustness.

2) Experiments on Testing Scenarios: To measure the robustness of $Best_{1h}$ and $Best_{2h}$, we evaluated and compared performance on 100 testing scenarios with different user distributions, three of which are shown in figure 4 and the rest, randomly generated. Table III compares GANet 1-hop and 2-hop versus ATRI performance with 156, 117, and 78 UAVs and different numbers of users on these testing scenarios. The table is segmented by the number of UAVs. The first two columns list the number of users and the deployment algorithm. The next three columns provide performance on the three test scenarios from figure 4. Since we can see the non-uniform user distributions for these scenarios in the figure, while the next subsequent column provides the average performance over **all** 100 test scenarios. This table shows that $Best_{2h}$ performed better on 95% (34 of 36) combinations of UAVs, users, and three testing scenarios, compared to $Best_{1h}$ and in all 36 compared to ATRI. These performance differences are also statistically significant with $p < 0.008$ for GANet 1-hop versus ATRI and $p < 0.00002$ for GANet 2-hop versus 1-hop. These results provide evidence of the robustness of GANet deployment. Once GANet optimizes potential field values for a set of UAVs on a set of training scenarios, the results indicate that we should expect similar network performance on new

TABLE III. COMPARING FITNESSES OF $Best_{1h}$, $Best_{2h}$, AND ATRI WITH DIFFERENT NUMBER UAVS AND USERS ON TESTING SCENARIOS.

Users	Methods	Te_1	Te_2	Te_3	Average	Te_1	Te_2	Te_3	Average	Te_1	Te_2	Te_3	Average
156 UAVs						117 UAVs				78 UAVs			
200	$ATRI$	1.54	1.47	1.43	1.6	1.69	1.37	1.57	1.48	1.48	1.28	1.32	1.19
200	1 hop	1.69	1.71	1.64	1.69	1.80	1.72	1.63	1.56	1.74	**1.67**	1.75	1.35
200	2 hop	**1.97**	**1.97**	**1.94**	**1.96**	**1.97**	**1.83**	**1.92**	**1.84**	**1.91**	1.65	**1.83**	**1.59**
150	$ATRI$	1.47	1.33	1.38	1.51	1.58	1.31	1.47	1.33	1.48	1.22	1.34	1.16
150	1 hop	1.61	1.69	1.54	1.63	1.73	1.71	1.66	1.49	1.83	**1.67**	1.72	1.30
150	2 hop	**1.98**	**1.99**	**1.97**	**1.97**	**1.98**	**1.84**	**1.96**	**1.85**	**1.9**	1.64	**1.86**	**1.58**
100	$ATRI$	1.41	1.31	1.31	1.39	1.50	1.09	1.40	1.23	1.50	1.05	1.40	1.1
100	1 hop	1.53	1.56	1.48	1.52	1.67	1.60	1.58	1.4	1.71	1.55	1.72	1.17
100	2 hop	**1.99**	**1.76**	**1.98**	**1.92**	**2**	**1.84**	**2**	**1.77**	**1.91**	**1.59**	**1.91**	**1.53**
50	$ATRI$	1.28	1.20	1.26	1.16	1.27	1.12	1.31	1.09	1.32	1.02	1.17	0.78
50	1 hop	1.33	1.35	1.39	1.33	1.47	1.34	1.43	1.3	1.62	1.32	1.56	1.01
50	2 hop	**1.96**	**1.87**	**1.91**	**1.59**	**1.59**	**1.56**	**1.97**	**1.52**	**1.84**	**1.36**	**1.78**	**1.37**

unseen scenarios with different numbers of users, UAVs, and user distributions.

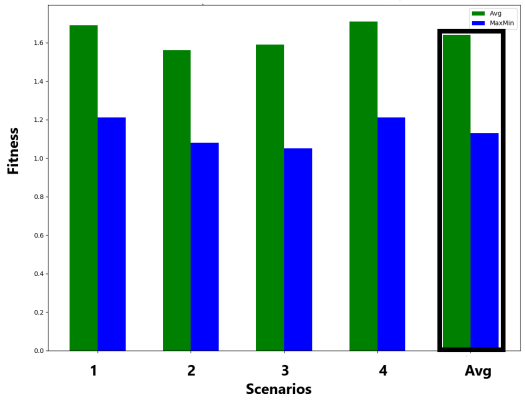


Fig. 13. GANet performance when maximizing the average performance across four training scenarios (green) compared to maximizing the minimum performance (blue).

3) Maximizing The Minimum Objective: In all previous experiments, we aimed to maximize fitness computed as the *average* performance (Equation 1) over four training scenarios. A common alternative is to maximize the *minimum* performance [39]. In experiments, comparing GANet 1-hop performance with maximizing the minimum (maximin) versus maximizing the average, we found that maximizing the average performs significantly better. Figure 13 shows that the best solution evolved using averaged fitness, green bar, performed better than the best solution evolved with maximin fitness, blue bar, on all four scenarios. On average, averaged fitness performed 45.13% better than maximin fitness. We conjecture that averaging performance better maintains diversity in the population and hence genetic algorithm explore more and thus leads to better performance.

VI. CONCLUSIONS

This paper uses genetic algorithm optimized potential fields to optimize UAV-based ad-hoc wireless network deployment. Unlike prior work, potential fields serve to unify representation across both search and service phases of network deployment and our proposed GANet optimizes potential field parameters that work robustly for different user distributions. We showed how to derive potential field parameters for the search phase and designed an elitist genetic algorithm to optimize non-linear potential field parameters for the service phase. We then formulated the service phase network deployment problem as a single objective optimization problem that maximizes the sum of bandwidth coverage and longevity of the deployed network. Performance is computed from running a simulation of UAVs moving under the influence of GANet optimized potential fields and provides a fitness metric for the genetic algorithm's search for optimal potential field parameter values. We showed that GANet is significantly better than gradient ascent and exhaustive search by finding a global optimum on a tractable problem more than twice as quickly as the nearest competitor. We then attacked larger, more realistic problems using four training scenarios with 156 UAVs and 200 users distributed uniformly and non-uniformly over an area of interest.

Results show that GANet network deployment is significantly better than the state of the art ATRI deployment and that GANet with two-hop UAV communications beats one-hop UAV communication performance. These results carry over to 100 test scenarios with never before seen user distributions, numbers, and to different numbers of UAVs. This provides evidence for GANet's robustness and our potential fields based representation. GANet adapts better when users are non uniformly distributed and matches or exceed ATRI performance with more uniform user distributions. Since with sufficient numbers of UAVs we get to optimal or near-optimal performance with two-hop UAV communication, we believe that going to higher hop counts for communications will not significantly improve performance.

REFERENCES

[1] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[2] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous uav surveillance in complex urban environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2, 2009, pp. 82–85.

[3] S. J. Lee, D. Lee, and H. J. Kim, "Cargo transportation strategy using t3-multirotor uav," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4168–4173.

[4] F. Nex and F. Remondino, "Uav for 3d mapping applications: a review," *Applied geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[5] M. Erdelj, M. Król, and E. Natalizio, "Wireless sensor networks and multi-uav systems for natural disaster management," *Computer Networks*, vol. 124, pp. 72–86, 2017.

[6] N. H. Motlagh, M. Bagaa, and T. Taleb, "Uav-based iot platform: A crowd surveillance use case," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 128–134, 2017.

[7] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, 2019.

[8] R. A. Nazib and S. Moh, "Energy-efficient and fast data collection in uav-aided wireless sensor networks for hilly terrains," *IEEE Access*, vol. 9, pp. 23 168–23 190, 2021.

[9] H. Li and A. V. Savkin, "Wireless sensor network based navigation of micro flying robots in the industrial internet of things," *IEEE Transactions on industrial informatics*, vol. 14, no. 8, pp. 3524–3533, 2018.

[10] D. Reina, H. Tawfik, and S. Toral, "Multi-subpopulation evolutionary algorithms for coverage deployment of uav-networks," *Ad Hoc Networks*, vol. 68, pp. 16–32, 2018.

[11] J. H. Holland, "Genetic algorithms," *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[12] M. Srinivas and L. M. Patnaik, "Genetic algorithms: A survey," *computer*, vol. 27, no. 6, pp. 17–26, 1994.

[13] M. Ma and Y. Yang, "Adaptive triangular deployment algorithm for unattended mobile sensor networks," *IEEE Transactions on Computers*, vol. 56, no. 7, pp. 946–847, 2007.

[14] Z. Gáspár and T. Tarnai, "Upper bound of density for packing of equal circles in special domains in the plane," *Periodica Polytechnica Civil Engineering*, vol. 44, no. 1, pp. 13–32, 2000.

[15] T. Chevet, C. S. Maniu, C. Vlad, and Y. Zhang, "Voronoi-based uavs formation deployment and reconfiguration using mpc techniques," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2018, pp. 9–14.

[16] M.-l. Lam and Y.-h. Liu, "Heterogeneous sensor network deployment using circle packings," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4442–4447.

[17] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of uav-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2016.

[18] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage," *IEEE Communications Letters*, vol. 20, no. 8, pp. 1647–1650, 2016.

[19] N. Bartolini, T. Calamoneri, T. F. La Porta, and S. Silvestri, "Autonomous deployment of heterogeneous mobile sensors," *IEEE Transactions on Mobile Computing*, vol. 10, no. 6, pp. 753–766, 2010.

[20] A. N. Patra, P. A. Regis, and S. Sengupta, "Distributed allocation and dynamic reassignment of channels in uav networks for wireless coverage," *Pervasive and Mobile Computing*, vol. 54, pp. 58–70, 2019.

[21] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[22] A. C. Woods and H. M. La, "A novel potential field controller for use on aerial robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 4, pp. 665–676, 2017.

[23] R. Dubey, J. Ghantous, S. Louis, and S. Liu, "Evolutionary multi-objective optimization of real-time strategy micro," in *2018 IEEE Conference on Computational Intelligence and Games (CIG)*. IEEE, 2018, pp. 1–8.

[24] R. Dubey, S. Louis, A. Gajurel, and S. Liu, "Comparing three approaches to micro in rts games," in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 777–784.

[25] A. Howard, M. J. Matarić, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem," in *Distributed Autonomous Robotic Systems 5*. Springer, 2002, pp. 299–308.

[26] S. Poduri and G. S. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 165–171.

[27] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for uav airborne networks toward on-demand coverage," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2015–2031, 2018.

[28] D. Reina, H. Tawfik, and S. Toral, "Multi-subpopulation evolutionary algorithms for coverage deployment of uav-networks," *Ad Hoc Networks*, vol. 68, pp. 16–32, 2018.

[29] D. S. Deif and Y. Gadallah, "Wireless sensor network deployment using a variable-length genetic algorithm," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, pp. 2450–2455.

[30] R. Jain, F. Templin, and K.-S. Yin, "Analysis of l-band digital aeronautical communication systems: L-dacs1 and l-dacs2," in *2011 Aerospace Conference*. IEEE, 2011, pp. 1–10.

[31] L. J. Eshelman, "The chc adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of genetic algorithms*. Elsevier, 1991, vol. 1, pp. 265–283.

[32] R. Dubey, S. J. Louis, and S. Sengupta, "Evolving dynamically reconfiguring uav-hosted mesh networks," in *2020 IEEE Congress on Evolutionary Computation (CEC)*, 2020, pp. 1–8.

[33] X. Tian, Y. Guo, R. R. Negenborn, L. Wei, N. M. Lin, and J. M. Maestre, "Multi-scenario model predictive control based on genetic algorithms for level regulation of open water systems under ensemble forecasts," *Water Resources Management*, vol. 33, no. 9, pp. 3025–3040, 2019.

[34] J. Hagelbäck and S. J. Johansson, "Using multi-agent potential fields in real-time strategy games," in *Seventh International Conference on Autonomous Agents and Multi-agent Systems (AAMAS), 12-16, 2008, Estoril*, 2008, pp. 631–638.

[35] 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*, vol. PP, no. 99, pp. 1–1, 2016.

[36] S. J. Louis and S. Liu, "Multi-objective evolution for 3d rts micro," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[37] S. J. Louis, T. Jiang, and S. Liu, "Real-time strategy game micro for tactical training simulations," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 2018, pp. 1656–1663.

[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[39] J. W. Herrmann, "A genetic algorithm for minimax optimization problems," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 2. IEEE, 1999, pp. 1099–1103.