Multi Objective UAV Network Deployment for Dynamic Fire Coverage

Kripash Shrestha, Rahul Dubey, Ashutosh Singandhupe, Sushil Louis, Hung La
Computer Science and Engineering
University of Nevada Reno, USA
(kshrestha, rdubey018, asingandhupe)@nevada.unr.edu, (sushil, hla)@unr.edu

2021 IEEE Congress on Evolutionary Computation (CEC) | 978-1-7281-8393-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/CEC45853.2021.9504947

***Abstract*—Recent large wildfires and subsequent damage have increased the importance of wildfire monitoring and tracking. However, human monitoring on the ground or in the air may be too dangerous and we thus investigate deploying Unmanned Aerial Vehicles (UAVs) to track wildfires. Specifically, we attack the problem of distributed autonomous control of UAVs using a set of potential fields to track wildfire boundaries. A multi-objective evolutionary algorithm searches through the space of potential field parameters to maximize fire coverage while minimizing energy consumption. Fire spread is modelled by the well known FARSITE fire model. Preliminary simulation results show that our potential fields approach to UAV control leads to 100% coverage of the boundary by UAVs and 78.1% energy remaining on three testing scenarios.**

Keywords—Distributed control, multi objective optimization, fire tracking, and UAV.

I. Introduction

Wildfires are spontaneous events that are known to cause massive destruction to structures and wildlife. The Congressional Research Service reported that for every year since 2000, there have been an average of $70,685$ wildfires that burned an average of 7.1 million acres in the United States alone [1]. Although on average, there were $78,600$ wildfires annually in the US in the 1990s, the total acres burned have more than doubled. Fighting wildfires is dangerous as the behavior of wildfires can be unpredictable and difficult to model. The National Fire Protection Association reports that from 2014 to 2018, on average 65 firefighters' lives have been lost annually while fighting wildfire [2] and these numbers do not even account for the number of non-firefighter civilian lives lost to wildfires. The loss of wildlife, human life, and structures highlights the importance of the need to locate, observe and track wildfires. This information is critical to making emergency plans to evacuate civilians to safety and to fight the fires.

UAVs can and have been used to assist humans in emergency and disaster situations by providing situational awareness with imagery and maps [3]. By maintaining proper communication links between the UAVs and ground control stations, we can remotely and safely assess damage in a given region of interest. Thus, UAVs are highly suitable for tackling the wildfire tracking problem by providing imagery and maps while relaying information through each other to firefighters/operators who are at a safe distance [4]. Teaming of UAVs and other robots to collaborate for resolving multiple challenges has risen in popularity for both research and application. Multiple UAVs have been used to collaborate as a network of agents with sensors to build maps and gather information in their local areas to get accurate information of the entire scope of the problem.

We use multiple UAVs to track the spread of a wildfire by monitoring fire fronts. This work takes inspiration from [4], [5] that used UAVs to deploy UAV-hosted ad-hoc wireless mesh networks and potential fields to control the movement of these UAVs. Formulating fire boundary tracking as a two-objective optimization problem, we use the same potential fields based approach as described in [5] to control these UAVs. Specifically, we aim to maximize fire boundary coverage and minimize the energy consumption of deployed UAVs in a network. The first objective ensures that we can track the fire spread while the second maximizes the length of time UAVs can stay in the air while tracking the fire. We then use our own implementation of Deb's NSGA-II [6] to optimize potential field parameters encoded in a real-valued chromosome. The tuned potential field parameters control a team of UAVs that maneuver autonomously to track the spread of the fire front in a distributed manner. Past work has shown that potential fields are suitable for distributed control tasks since, once tuned, they have low computational requirements, can be visualized to help explain control behavior, and have been used to produce complex task achieving behavior for heterogeneous groups of agents [5], [7], [8]. However, being highly non-linear makes them difficult to optimize leading to our use of an evolutionary algorithm for optimizing potential field parameters to our fire tracking task.

In this work, we are interested in tracking fire *boundaries* and not the entire area covered by a fire and we therefore wish to position UAVs along fire boundaries. Fires can spread quickly and unpredictably in any direction, thus we simulate different fire scenarios using the well-known FARSITE fire simulator [9]. We evaluate an individual in the evolutionary algorithm's population in simulation. The individual's potential field parameters determine UAV movement and we simulate both fire spread and UAV movement for a set number of time steps. At the end, we compute fire boundary coverage and remaining battery energy. Our experiments use three different simulated scenarios to evaluate the fitness of an individual and we average coverage and energy values over all three when evaluating each individual in the population. Once the evolutionary algorithm has tuned the parameters on these three scenarios, we test robustness of tuned parameter values on three more hitherto unseen scenarios and report results. Preliminary results show that on both training and testing scenarios we achieve 100% fire boundary coverage with 72% and 78% remaining energy at the end of simulations on training and testing scenarios respectively. These results show the strong potential for potential fields based distributed control and also provide evidence for the generalizability of such an approach to different domains [5].

The main contributions of the paper are twofold. First, we provide a new approach to distributed autonomy for fire boundary tracking using autonomous UAVs controlled by potential fields. Second, we show that an evolutionary multi-objective algorithm provides near optimal tuning of the highly non-linear potential field parameters leading to robust autonomous fire boundary tracking.

The remainder of the paper is organized as follows. Section II describes prior work in using UAVs for wildfire tracking. Section III sets up the problem and provides experimental details. Section IV discusses evolutionary distributed control, the fire influence map, and our fitness computation. Section V presents experimental results on our three training and three testing scenarios. Finally, Section VI summarizes the findings of the paper and suggests possible future work.

II. Background

Although UAVs find many applications in both civilian and military domains [10], [11], we focus on using UAVs for tracking the spread of wildfires in forests. Two fundamental challenges when using UAVs in unknown regions for fire coverage are fire detection and fire tracking and this paper focuses on the latter. However, the two are related as we need detection for tracking, so we start with providing an overview of work in detection. Yuan [12] used Infrared (IR) imaging sensors installed on UAVs to detect the presence of fire and presented techniques to process images gathered using different sensors mounted on UAVs, to study fire spreading behaviors. Afghah [13] proposed a leader-follower formation to cluster a set of UAVs into multiple coalitions that collectively covered a particular area of interest. Merino [14] proposed a cooperative perception system for multiple heterogeneous UAVs for automatic detection of forest fires. They collected data using multiple sensors such as a visual cameras, infrared sensors, and fire detectors mounted on UAV's and fused them together for detection, monitoring, and measurement of forest fires. Another fire detection technique developed by Yuan [15] analyzes fire segmentation in different color spaces. Henry [16] introduced a Forest Fire Detection Index (FFDI) to detect fires through the use of a new color index. The index is based on method for vegetation classification and used to detect flames and smoke.

The work in fire detection has shown that several robust techniques exist for fire detection and tracking as long as we have good observation platforms with suitable sensors. In this paper, we assume that UAVs have a fire detection sensor and our challenge is to find the fire boundary and move UAVs to track this boundary as it grows in perimeter length over time. Coordinated control of multiple UAVs is essential for dynamic fire coverage and control techniques are broadly categorized as either distributed or centralized control. David [17] presented a path planning algorithm to track fire using low altitude short endurance UAVs. This centralized path planning computes waypoints for each UAV, with these waypoints being generated along the edge of a fire (along the fire boundary). Phan [18] worked on a similar problem where he proposed a cooperative control framework for a team of UAVs and unmanned ground vehicles (UGVs) to detect and track fires. In this centralized framework, a mission controller monitors a dynamic environment, formulates high level mission plans, and allocates task to each vehicle.

Our potential field based approach is a distributed control approach where each UAV collects information about the Area of Interest (AOI) using mounted sensors and communicates with neighbors to make independent decisions in real-time. Many researchers have studied distributed control techniques for UAVs. Manish [19] investigated cooperative control of multiple UAVs for accurate situational awareness and distribution of fire suppressant fluid at the edges of fire. Maza [20] proposed a distributed decision making architectural framework for multi-UAV configuration in disaster management. Multiple checkpoints in the region of interest can be used to command UAVs to track down the boundary for fire coverage. The authors of [21] presented a deep learning approach to deploy UAVs to collect images for fire classification.

Open challenges still include the coverage of boundaries of a fire when the fire spreads dynamically. Earlier approaches to resolve this problem include [17] and [18] where UAV maneuvering decision making was centralized. Centralized approaches suffer from a single point of failure and may need more computation and communication hardware than available. However, given enough computational and communication resources, these approaches can direct a team of UAVs to continuously track the spread of fire along the boundary as long we have a good fire spread model or external information about fire boundary. Simultaneous detection and tracking in a decentralized manner remains challenging. Closest to the work in this paper, Hung La used a team of UAVs working together to track and follow a wildfire as it spreads by tracking fire intensity and heat sources [22]. However, the only objective was to maximize fire boundary coverage and there was no modeling of energy usage. Our work describes a new control approach and formulates the problem as a two objective optimization problem of maximizing coverage while minimizing energy consumption.

III. Problem Statement

In this work, we aim to deploy a set of UAVs to track wildfire spread in unknown environments. Let us assume that we have a set of n homogeneous UAVs $u = \{u_1, u_2...., u_n\}$ and a rectangular Area of Interest (AOI). Here homogeneous refers to UAVs with same attributes as shown in Table I. We assume that the fire is non-uniformly distributed over the area of interest and that each UAV is equipped with GPS devices and downward-facing cameras capable of detecting fire. Each camera has a circular Field of View (FOV) and can capture the area under its FOV. Images captured by each UAV can be sent back to a Command Center (CC) for analysis so we can locate areas on fire. For experimentation, we simulate a fire model called FARSITE [22] and used different FARSITE parameters to get different behaviors of fire spread. Our objective is to track fire spread for as long as possible and we thus use minimizing energy consumption as an optimization objective in our multi objective optimization problem formulation.

Let us assume that the k^{th} UAV observes I_k fire intensity under the UAV and has E_k remaining energy at the end of a simulation. Since we want to maximize both fire boundary coverage and the amount of time we maintain coverage, the problem can be formulated as a two-objective optimization problem described by equation 1. The first term of the equation refers to boundary coverage and here we assume fire intensity coverage is proportional to the boundary coverage. The entire boundary is covered when $\sum I_k = 100\%$. More specifically.

$$\text{Maximize} f = \left[\frac{\sum_{k=1}^{n} I_k}{I_{max}}, \frac{\sum_{k=1}^{n} E_k}{E_{max}} \right] \tag{1}$$

Here I_{max} is maximum intensity at the boundary and E_{max} is the sum of energy over all UAVs at simulation start. n is the number of UAVs and dividing by I_{max} and E_{max} normalizes objective values to lie between 0 and 1.

IV. METHODOLOGY

We use a set of potential fields to govern the movement of UAVs to track a fire's boundary and start by describing the fire simulator that models fire boundary movement. The Fire Area Simulator (FARSITE) model [9] is a well-known core model of existing fire simulation systems. Apart from FARSITE, other fire simulation models [23] can be used to simulate fire propagation but FARSITE seems the most popular, reliable model and is widely used by federal land management agencies such as the US Department of Agriculture [9]. FARSITE estimates fire fronts based on an elliptical model introduced by Richards [24] and requires and uses information on terrain, geography, topography, fuel, and weather to model the spatial distribution and spread of a fire. Since accurate fire simulation is not in the scope of this paper, we use simplified fire propagation in FARSITE to calculate fire front growth [22].

We assume that fire initially starts with 33 heat sources, $H = \{(x_1, z_1), (x_2, z_2), ..., (x_{33}, z_{33})\}$ on the xz plane where x_1, z_1 are the x and z coordinates of the 1^{st} heat source location. Figure 1(a) shows the initial heat source locations on a grid with axes units in meters. Each heat source propagates in the direction of the wind using equation 2 where x_t and z_t are the differentials of the fire on the xz plane and θ, $(0 \leq \theta \leq 2\pi)$, is the direction of the wind. c is the distance from the heat source to the center of the ellipse which can be empirically calculated [9].

$$x_t = c * \cos(\theta), z_t = c * \sin(\theta) \tag{2}$$

$$\text{c} = \frac{R - \frac{R}{HB}}{2}$$

$$\text{HB} = \frac{(LB + (LB^2 - 1)^{0.5})}{(LB - (LB^2 - 1)^{0.5})} \tag{3}$$

$$\text{LB} = 0.936e^{0.2566U} + 0.461e^{-0.1548U} - 0.397$$

Equation 3 computes c which depends on two variables, R, the rate of fire spread and U, the wind speed. For experiments, we used $25m/s$ for the rate of fire spread, and $5m/s$ for wind speed. The intensity of the fire point location is initially set to random values from a normal distribution ranging from $2-10$ kW/m which remains constant throughout the propagation of fire. We compute the location of the new fire front after every Δt and equation 4 computes the next propagation location of the i^{th} heat source.

$$\begin{aligned} x_i(t + \Delta t) &= x_i(t) + \Delta t X_t(t) \\ z_i(t + \Delta t) &= z_i(t) + \Delta t Z_t(t) \end{aligned} \tag{4}$$

We ran FARSITE for 6000 time steps for one evaluation.Since we are interested in covering the fire boundary, not in all locations covered by fire, we extract the boundary using a fire influence map.

A. Fire Influence Map

Algorithm 1: Fire Influence Map

Input : Initial heat source locations $H = \{(x_1, z_1), (x_2, z_2), ..., (x_h, z_h)\}$
Output: Grid

for *t in maxtimeSteps* **do**
  for *h in H* **do**
    $\text{x}_h(t + \Delta t) = x_h(t) + \Delta t x_h(t)$
    $\text{z}_h(t + \Delta t) = z_h(t) + \Delta t z_h(t)$
    Intensity$_h$ =random(2,10)
    for i *in* $Grid_x$ **do**
      for j *in* $Grid_z$ **do**
        D = Distance($Grid[i][j], (\text{x}_h(t), z_h(t))$)
        if *D == 0* **then**
          Grid[i][j] += Intensity$_h$
        end
      end
    end
    H.append(($\text{x}_h(t + \Delta t), z_h(t + \Delta t)$))
  end
end
return(Grid)

A Fire Influence Map (FIM) is a grid defining fire spatial information in a given AOI with values assigned to each grid-cell based on fire intensity. Algorithm 1 computes these values for each grid cell. Algorithm 1 tracks the fire starting with initial heat source locations and runs for 6000 time-steps. At each time step, we calculate fire intensities and the spread of the fire using equation 4 Figure 1(a) shows the initial 33 heat sources and Figure 1(b) shows the fire spread at time step

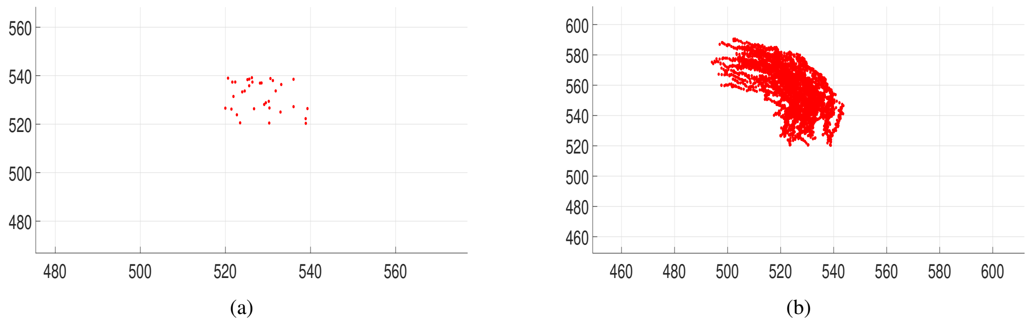


Fig. 1. A discretized fire influence map. (a) 33 initial fire heat sources at $t = 0.$, and (b) Propagation of fire from initial 33 heat sources at time step $t = 100$.

100. We extract the fire boundary using grid values at the end of each time step, and a team of UAVs maneuver to cover this extracted boundary. The boundary is a set of tuples $B = \{(x_1, z_1, i_1), (x_2, z_2, i_2), ..., (x_k, z_k, i_k)\}$ where x_k, z_k are the x and z coordinates of k^{th} fire location with fire intensity of i_k on the boundary.

In the literature many boundary extraction techniques such as square tracking, moore-neighbor tracing, and radial sweep exist [25], [26]. For simplicity, we implemented our own version of radial sweep to extract fire fronts. We scan the FIM grid, first, from left to right, and second, from top to bottom, storing the first and last grid locations with positive fire intensity values as the boundary. This gives us the fire boundary and we next describe how we deploy UAVs to track this boundary using potential fields.

B. UAV Deployment

UAV based deployment typically occurs in two phases [5]. In the first phase, UAVs are deployed in a fixed or static topology with the aim of covering the entire area in a given AOI in order to find items of interest - in this case, fire. We use a Circle Packing Theorem (CPT) based algorithm for initial positioning of UAVs at positions given by the CPT and at a starting elevation of 100 meters [27]. Althought there are other techniques such as Delauney Triangulation (DT) [28] and Voronoi Diagrams (VD) [29] that can be used to deploy UAVs in the first phase, we use CPT because it is a fast, simple technique and ensures no overlapping between UAVs' coverage radii while maximizing area covered. During this first phase, each UAV identifies locations on fire within its sensor range and communicates this information to neighbors. Only in the second phase do UAVs start moving under the influence of different attractive and repulsive potential fields.

C. Potential Field and Representation

Potential Field (PF) based real-time movement control of agents was first introduced by Khatib in 1986 [30]. This became a popular collision avoidance technique owing to its simplicity and wide applicability. The original idea was to use distance based potential fields to avoid colliding with static and dynamic obstacles while moving towards a goal location. In this paper, we augment distance based potential fields with attractive and repulsive potential fields based on fire intensity to control the movement of UAVs. Equation 5 shows the resultant potential fields equation comprised of three distance dependent potential fields and two fire intensity dependent potential fields that govern the heading of the k^{th} UAV. We explain each term in the equation below.

We use potential fields based on distance to other UAVs, distance to the fire boundary, intensity of fire, UAV altitude, and neighboring UAV's fire coverage. PF^k_{fb} denotes one attractive and one repulsive potential field dependent on the distance from the k^{th} UAV to the nearest fire boundary location so UAVs are attracted towards the boundary but do not get too close to the fire itself. PF^k_{fI} similarly defines the two potential fields associated with fire intensity and also helps influence coverage, while PF^k_{nd} denotes potential fields based on inter UAV distance so that UAVs do not cluster too close to each other. PF^k_{alt} mediates altitude and coverage since height correlates with energy usage and with coverage FOV. The last set of potential fields, PF^k_{nI}, are based on neighbors fire coverage.

Each term in equation 5 denotes two potential fields with each potential field having two parameters. The full set of equations with all potential fields and their parameters are show in in equations 6 to 10 respectively. Equation 6 shows the attractive and repulsive potential fields based on distance between the k^{th} UAV and its p neighbors within air-to-air communication range. Equation 7 computes potential fields similar to equation 6 except potential fields are dependent on neighbors fire intensity coverage. Equation 8 and 9 compute attractive and repulsive potential fields based on distance and fire intensity of m nearest fire points on the boundary of the fire. As mentioned earlier, the coverage radius of a UAV is directly proportional to its altitude and thus with more altitude, the coverage radius and area increase. However, a UAV expends more energy to change altitude compared to moving at constant altitude or hovering in place. Thus we generate an attractive and a repulsive potential field using equation 10 to control the altitude of UAVs and find a good trade-off between altitude and energy consumption.

$$UAV^k_{pf} = PF^k_{nd} + PF^k_{nI} + PF^k_{fb} + PF^k_{fI} + PF^k_{alt} \quad (5)$$

$$PF^k_{nd} = \sum_{j \in p} (c_1 d^{e_1}_{kj} - c_2 d^{e_2}_{kj}) \quad (6)$$

$$PF_{nI}^{k} = \sum_{j \in p} (c_3 n I_{kj}^{e_3} - c_4 n I_{kj}^{e_4}) \tag{7}$$

$$PF_{fb}^{k} = \sum_{j \in m} (c_5 d_{kj}^{e_5} - c_6 d_{kj}^{e_6}) \tag{8}$$

$$PF_{fI}^{k} = \sum_{j \in m} (c_7 f I_{kj}^{e_7} - c_8 f I_{kj}^{e_8}) \tag{9}$$

$$PF_{alt}^{k} = c_9 d_k^{e_9} - c_{10} d_k^{e_{10}} \tag{10}$$

Since each potential field has two parameters, a coefficient (c) and an exponent (e) to tune, we thus have 20 total parameters needing to be tuned. We encode these parameters in a real-value chromosomes with upper and lower bounds of coefficients $c \in [-500000, 500000]$ and exponents $e \in [-2, 2]$. These values were experimentally determined to work well and led to high fitness solutions.

Algorithm 2: Fitness Computation

Input : B, CS, MS, MT, UAVs
Output: fitness

```
obj1 = obj2 = 0;
for scenario in MS do
    timeSteps = 0;
    while timeSteps<MT do
        AssociateBoundary(B, UAVs);
        FindNeighbors(UAVs);
        Headings =
          ComputePotentialFields(CS, B, UAVs);
        MoveAll(Headings);
        timeSteps++;
    end
    obj1 += FireCoverage();
    obj2 += RemainingEnergy();
end
obj1 = obj1 / MS;
obj2 = obj2 / MS;
fitness = [obj1, obj2];
return(fitness);
```

D. Fitness Computation

The fitness of each chromosome or candidate solution is evaluated in three steps. First, we send potential field parameters to our simulator running one fire scenario. We then load these parameters to the UAV controller to control UAV movement and run the simulation for 6000 time steps. When finished, we compute the two objective values, boundary coverage and energy consumption for this scenario. We have three training scenarios so we repeat these steps for the remaining scenarios and return the average objective values over these three scenarios as the candidate solution's evaluation. Algorithm 2 specifies this fitness computation. Note that prior work indicates that solutions evolved on just one scenario may not produce similar results on other unseen scenarios [5], [7]. Thus, to evolve more robust parameter values, we used three different scenarios to evaluate each candidate solution and averaged results.

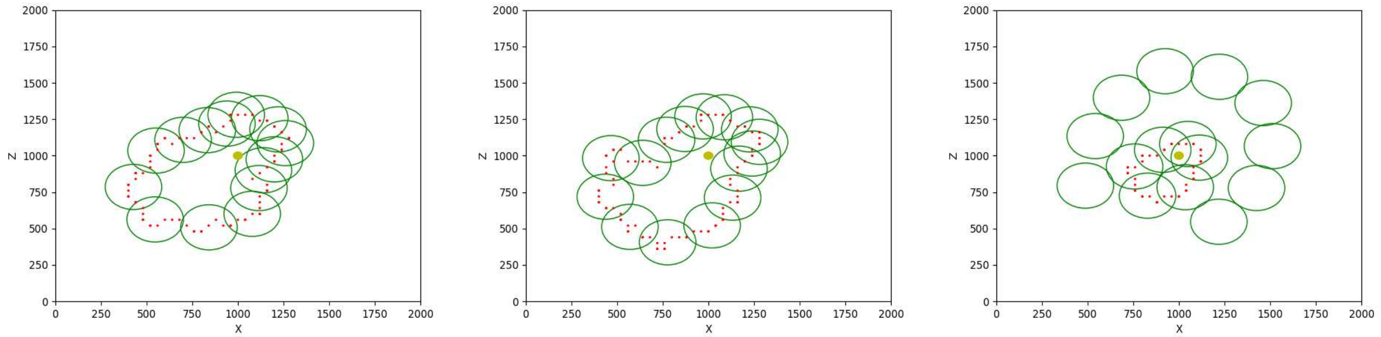
TABLE I. SIMULATION PARAMETERS.

	Parameters	Symbol	Value
UAV	Area of Interest	AOI	$2000m^2$
	Coverage Radius	U_{gr}	h
	A-2-A Range	U_{ar}	300m
	Max Speed	U_s	5 m/s
	Initial energy	E_t	10^6 J
	Hovering Energy	E_h	138.24 J/t
	Altitude Energy	E_h	164.16 J/t
	Linear Energy	E_h	109.44 J/t
NSGA-II	Population size	Pop_m	20
	Max generation	Gen_m	20
	Crossover rate	P_x	0.95
	Mutation rate	P_m	0.05
FARSITE	Wind Direction	θ	$0 \leq \theta \leq 2\pi$
	Wind Speed	U	$5\ m/sec$
	Fire Spread Rate	R	$25\ m/min$
	Intensity Values	I_{jt}	$(1-10)\ kW/m$

Algorithm 2 takes fire boundary (B), a candidate solution (CS), the number of scenarios (MS), a number of time steps (MT), and a set of UAVs, and returns the averaged fitness assigned to the CS. `AssociateBoundary` allocate a region of fire boundary to the closest UAV based on distance and `FindNeighbors` assigns neighboring UAVs based on air-to-air communication range. Using this boundary and neighbor information, `ComputePotentialFields` computes the vector sum of potential fields acting on this UAV. These *vector sum directions* are used by `MoveAll` to move each UAV in the direction indicated by the vector sum of potential fields acting on the UAV. At the end of the 6000 time steps for each scenario, two objective values are computed using equations 1. The first objective, Fire Coverage (FC), computed by equation 11 is the sum of the fire intensity covered by the UAVs divided by the sum of the total fire intensity at the fire boundary at the end of the simulation. We do this to normalize the objective value to be between zero to one. If there is an overlap of coverage by two or more UAVs where they cover the same segment of boundary, the intensity will only be counted towards this objective once. The second objective, Remaining Energy (RE), calculated by equation 12 is the sum of the remaining battery life of the UAVs divided by the total number of UAVs. UAV battery life ranges between zero and one with one representing a full charge and zero representing an empty battery. The two objective values are averaged over the three scenarios and assigned as fitness. In equation 11, I_i is the fire coverage of the i^{th} UAV and I_{max} is the total fire intensity at the boundary. Similarly, in equation 12, E_i is the remaining energy of i^{th} UAV and E_{max} is the sum of energy of all UAVs at the beginning. Table I provides UAV, evolutionary algorithm, and fire simulation parameters.

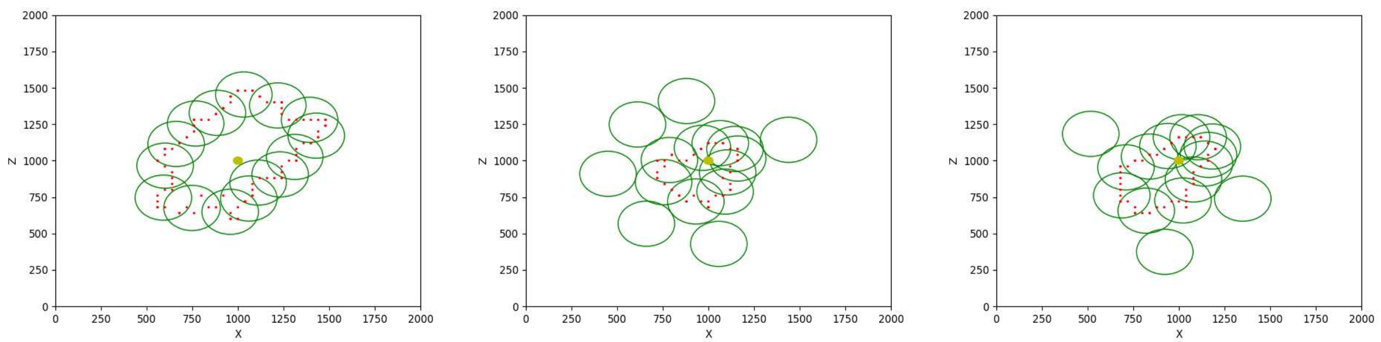
$$FC = \frac{\sum_{i=1}^{n} I_i}{I_{max}} \tag{11}$$

$$RE = \frac{\sum_{i=1}^{n} E_i}{E_{max}} \tag{12}$$



(a) Tr_1 (b) Tr_2 (c) Tr_3

Fig. 2. Fire tracking over three training scenarios where red dots represent the fire boundary in a given AOI. Green circles represent the coverage range of UAVs. In all three cases, 100% fire boundary coverage is achieved.



(a) Te_1 (b) Te_2 (c) Te_3

Fig. 3. Fire tracking over three testing scenarios. In all three cases, 100% fire boundary coverage is achieved.

V. Results And Discussion

We considered an area of interest of 2000 meters^2 and assume an initial 33 heat sources from which the fire starts and spreads. Each heat source starts near location $(1000, 0, 1000)$ in the xz plane. We used 15 homogeneous UAVs with circular fields of view. The coverage radius of each UAV is directly proportional to its altitude where minimum and maximum altitudes are restricted to be between 100 to 150 meters. Initially each UAVs starts at an altitude of 100 meters.

A. Training Scenarios

As mentioned earlier, solutions evolved on only one scenario may not produce good results on other unseen scenarios. In other words, evolved solutions may not be robust. Thus we evaluate each candidate solution on three different fire scenarios as shown in Figure 2. We obtained these different fire scenarios by changing variables in our FARSITE fire model. At the end of evaluation, objective values averaged over these three training scenarios are assigned as two objective values of a given candidate solution. The NSGA-II evolves the population and quickly produces good potential field parameter values. Figures 2 and 3 show UAV positions at the end of the simulation for one of the more balanced solutions on the pareto front. Specifically, we picked a solution from the last generation pareto front with objective values of $(1, 0.72)$ and Figure 2 shows the locations of the the fire boundary and the locations of UAVs at the end of simulation for the three training scenarios. Figure 2 shows that UAVs are able to cover 100% of the fire boundary in all three training scenarios, with 82.2%, 69.9%, 66.8% energy remaining respectively. Circles indicate coverage and the overlapping circles show that we have more than enough UAVs to cover the entire boundary within our altitude (and energy) constraints. Even with a relatively small population of size 20, the NSGA is able to quickly evolve good solutions.

B. Testing Scenarios

In order to evaluate the robustness of evolved solutions, we picked the same solution with objective values of $(1, 0.72)$ from the last generation pareto front and ran this solution on three unseen fire scenarios as shown in Figure 3. On the first testing scenario Te_1, at the end of 6000 time steps simulation we achieve 100% percentage fire coverage at the boundary with 79.7% energy still remaining. Similar running on the second and third testing scenarios achieved 100% fire boundary coverage and 77.5%, 79.7% remaining energy respectively. On average we achieved 78.1% remaining energy. These preliminary results indicate the viability of our potential fields based approach towards a decentralized control of UAVs performing a fire tracking task. However, much work remains in scaling our results and in establishing a firm theoretical foundation to provide performance and reliability limits.

C. Performance With Reduced Number of UAVs

We are now considering fewer UAVS since given enough UAVs, it is not surprising that we are able to achieve 100% coverage with significant energy to spare. We reduced the total number of available UAVs by 50% $(7.5 \approx 8)$ and using the same parameters from earlier $(1, 0.72)$ evaluated performance on all six training and testing scenarios. On average we get only about 88% fire coverage, 76.3% remaining energy, and Figure 4 shows that we still have some overlap. We need to better tune potential fields to minimize overlap and we may need to modify altitude limits. Evolving on training scenarios with different numbers of UAVs and explicitly including overlap in the objective function seem promising avenues for future research.

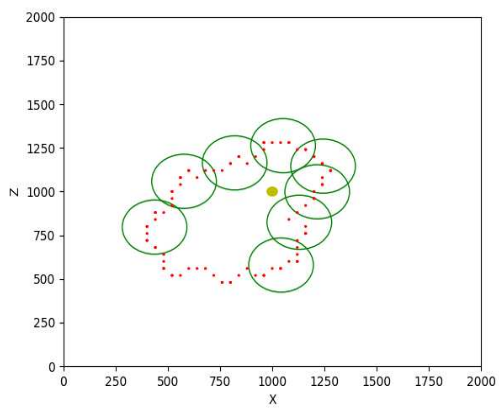


Fig. 4. Fire tracking with 8 UAVs.

VI. Conclusions And Future Work

In this paper, we proposed and evaluated a decentralized approach to UAV based fire boundary tracking. We formulated this problem as a multi-objective optimization problem that maximizes fire coverage while minimizing energy consumption. Our approach uses a set of potential fields to govern the movement of UAVs and we used an evolutionary algorithm to optimize parameter values to achieve objectives. Preliminary results indicate that potential fields based solutions show significant promise on this task and that an evolutionary algorithm evolves robust solutions that degrade gracefully in unseen environments. When trained with 15 UAVs, the NSGA evolves solutions with 100% fire boundary coverage even on unseen scenarios but the same parameter values lead to lower coverage when using 8 UAVs. More specifically, evolved solution performance on different training and testing scenarios show that given enough UAVs (15), we can get 100% fire boundary coverage while having 74.5% remaining energy at the end of our simulation runs. When we reduce the number of UAVs by half we still achieved an average of 88% boundary coverage with 76.3% energy reserves.

In the future, we plan to extend this work by varying the number of UAVs during training, considering more realistic physics to govern UAVs movement, rectangular field of view instead of circular, and incorporating machine learning techniques to analyze collected imaginary data using on-board cameras. Also, we plan to extend this work as a many objective optimization problem to incorporate more objectives.

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] "Wildfire statistics- congressional research service," https://fas.org/sgp/crs/misc/IF10244.pdf, last accessed: February 21, 2021.

[2] "Firefighter fatalities in the united states," https://www.nfpa.org/News-and-Research/Data-research-and-tools/Emergency-Responders/Firefighter-fatalities-in-the-United-States, last accessed: February 21, 2021.

[3] 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.

[4] H. X. Pham, H. M. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6648–6653.

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

[6] 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.

[7] 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)*, 2018, pp. 1–8.

[8] 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.

[9] M. Finney, "Farsite : Fire area simulator : model development and evaluation," 1998.

[10] D. Orfanus, E. P. de Freitas, and F. Eliassen, "Self-organization as a supporting paradigm for military uav relay networks," *IEEE Communications Letters*, vol. 20, no. 4, pp. 804–807, 2016.

[11] Y. Mualla, A. Najjar, A. Daoud, S. Galland, C. Nicolle, E. Shakshuki *et al.*, "Agent-based simulation of unmanned aerial vehicles in civilian applications: A systematic literature review and research directions," *Future Generation Computer Systems*, vol. 100, pp. 344–364, 2019.

[12] C. Yuan, Z. Liu, and Y. Zhang, "Fire detection using infrared images for uav-based forest fire surveillance," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2017, pp. 567–572.

[13] F. Afghah, A. Razi, J. Chakareski, and J. Ashdown, "Wildfire monitoring in remote areas using autonomous unmanned aerial vehicles," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2019, pp. 835–840.

[14] L. Merino, F. Caballero, J. R. Martinez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple uavs: Application to automatic detection of forest fires," *J. Field Robotics*, vol. 23, pp. 165–184, 03 2006.

[15] C. Yuan, Z. Liu, and Y. Zhang, "Uav-based forest fire detection and tracking using image processing techniques," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2015, pp. 639–643.

[16] H. Cruz, M. Eckert, J. Meneses, and J. Martínez, "Efficient forest fire detection index for application in unmanned aerial systems (uass)," *Sensors (Basel, Switzerland)*, vol. 16, 2016.

[17] D. Casbeer, R. W. Beard, T. Mclain, S. Li, and R. Mehra, "Forest fire monitoring with multiple small uavs," *Proceedings of the 2005, American Control Conference, 2005.*, pp. 3530–3535 vol. 5, 2005.

[18] C. Phan and H. H. T. Liu, "A cooperative uav/ugv platform for wildfire detection and fighting," in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, 2008, pp. 494–498.

[19] M. Kumar, K. Cohen, and B. HomChaudhuri, "Cooperative control of multiple uninhabited aerial vehicles for monitoring and fighting wildfires," *Journal of Aerospace Computing, Information and Communication*, vol. 8, pp. 1–16, 01 2011.

[20] I. Maza, F. Caballero, J. Capitan, J. R. Martinez-de Dios, and A. Ollero, "Experimental results in multi-uav coordination for disaster management and civil security applications," *Journal of Intelligent and Robotic Systems*, vol. 61, pp. 563–585, 01 2011.

[21] C. Kyrkou and T. Theocharides, "Emergencynet: Efficient aerial image classification for drone-based emergency monitoring using atrous convolutional feature fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 1687–1699, 2020.

[22] H. Pham, H. La, D. Feil-Seifer, and M. Deans, "A distributed control framework for a team of unmanned aerial vehicles for dynamic wildfire tracking," 09 2017.

[23] D. Albright and B. Meisner, "Classification of fire simulation systems," *Fire management notes*, 1999.

[24] G. D. Richards, "An elliptical growth model of forest fire fronts and its numerical solution," *International Journal for Numerical Methods in Engineering*, vol. 30, pp. 1163–1179, 1990.

[25] G. Toussaint, "Grids, connectivity, and contour-tracing," *URL:¡ http://www-cgrl. cs. mcgill. ca/˜ godfried/teaching/pr-notes/contour. ps*, 1988.

[26] M. Alder, "An introduction to pattern recognition," *Mike Alder*, 2001.

[27] 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.

[28] 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.

[29] G. Wang, G. Cao, and T. F. La Porta, "Movement-assisted sensor deployment," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.

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