Optimal UAV Positioning for a Temporary Network Using an Iterative Genetic Algorithm

Nicholas Ceccarelli
Computer Science & Engineering
University at Buffalo, The State University of New York
Buffalo, NY USA
njceccar@buffalo.edu

Paulo Alexandre Regis, Shamik Sengupta, David Feil-Seifer
Computer Science & Engineering
University of Nevada, Reno
Reno, NV USA
pregis@nevada.unr.edu, ssengupta@unr.edu, dave@cse.unr.edu

***Abstract*—Efficient arrangement of UAVs in a swarm formation is essential to the functioning of the swarm as a temporary communication network. Such a network could assist in search and rescue efforts by providing first responders with a means of communication. We propose a user-friendly and effective system for calculating and visualizing an optimal layout of UAVs. An initial calculation to gather parameter information is followed by the proposed algorithm that generates an optimal solution. A visualization is displayed in an easy-to-comprehend manner after the proposed iterative genetic algorithm finds an optimal solution. The proposed system runs iteratively, adding UAV at each intermediate conclusion, until a solution is found. Information is passed between runs of the iterative genetic algorithm to reduce runtime and complexity. The results from testing show that the proposed algorithm yields optimal solutions more frequently than the k-means clustering algorithm. This system finds an optimal solution 80% of the time while k-means clustering is unable to find a solution when presented with a complex problem.**

***Index Terms*—UAV network, genetic algorithm, positioning**

I. Introduction

UAVs (Unmanned Aerial Vehicles), colloquially known as drones, have seemingly endless use cases and applications. The ability to use a swarm of UAVs to complete a task is of particular interest. UAV swarms can perform countless functions, such as creating a temporary communication network for search and rescuers. Networks of a singular UAV have been created, though such a network would likely be connected to some other external network [1]. Whether the UAVs create a self-contained network (for short range communication) or are externally connected (for external communication such as cellular) [2], interconnected UAV base stations (UAV-BS) have useful applications. UAVs can support pre-existing infrastructure or create a new, standalone network. UAV-BS may play a crucial role in forthcoming technological advancements, such as the evolution of cellular connectivity [3]. Natural disasters can destroy existing communication infrastructure and render it useless [4]. Search and rescue could be improved by a standalone communication network in natural disaster areas or areas lacking infrastructure.

In creating a temporary network of UAVs, an optimized organization, rather than random placement, allows more network users to be connected simultaneously. Such a system will drastically improve search and rescue efforts in remote areas, and do so in a cost effective manner. We assume the UAVs form a mesh network for connectivity and one or more are designated to act as the gateway, similar to the system described in [3]. Fig. 1 illustrates an example architecture of such a system.

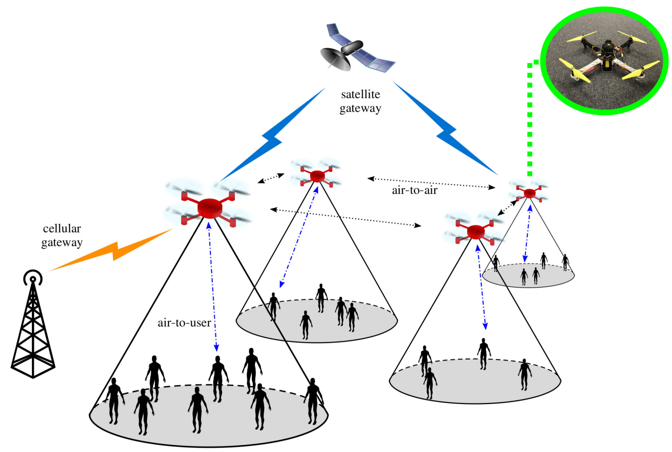

Fig. 1: Example of temporary network application

We propose an iterative genetic algorithm (GA) that computes the most efficient coverage for a subset of or all users in a given map to provide an optimized layout for the UAVs to arrange. The algorithm passes information from iteration-to-iteration, thus improving the solution at each step. Visualization software is also provided to display the arrangement of UAVs and search area over the original map.

Previous research has been consulted to calculate the coverage range of an antenna installed on a UAV. The antennas are pointed downward to maximize the signal projected onto the ground. The coverage range is a function of the height of the UAV and the beamwidth of the installed antenna [5]. The value for the beamwidth is dependent only upon the given specifications of an antenna, while the maximum height of the UAV can be calculated by using specifications of the antenna and Friis transmission equation [6].

Our proposed algorithm accepts a map of an area and a minimum percentage of the network users in the map that must be provided coverage. The map allows for areas of varying network demand. The higher the demand in a given area, the more likely that area will be provided coverage. The proposed algorithm can provide a map of the area with a minimal number of UAVs overlayed by using the predetermined and calculated parameters.

An iterative approach is used because the underlying GA alone is unable to account for multiple UAVs. Because of this limitation, the proposed algorithm runs the GA iteratively with one UAV, and if it fails, it attempts two, and so on until a solution is found. This alone would be computationally exorbitant, so information is passed between executions of the GA to provide increased efficiency.

The remainder of this paper is organized as follows: Section II discusses background information needed to fully understand the problem and presented solution, Section III describes the proposed algorithm, Section IV explains how the system has been validated and compares its performance to another leading algorithm: k-means clustering, and Section V lays out conclusions and future work possible in this field.

II. Background

UAVs can be organized into a swarm into a temporary network in the steps outlined in the following subsections.

A. Optimal UAV Positioning

Optimal positioning is ideal when using a UAV swarm to provide a means of communication. In the past, such swarms have been used for autonomous vehicles to communicate with one another [7]. This system is focused on UAV and network capabilities. The vehicles were theoretically provided with internet connectivity by overhead UAVs connected to a nearby base station. This system could be combined with the proposed algorithm to create a usable positioning scheme for UAVs.

A purely mathematical approach was used to find the optimal placement for a singular UAV [1], but it has not been applied to a multiple UAV system. Line-of-sight calculations were used to find the best location. An additional mathematical approach has been used to calculate optimal base station location, given an arrangement of UAVs [8]. This system requires extensive mathematical knowledge because complex calculations need to be performed for each UAV and each base station. The proposed algorithm does not require such comprehensive understanding.

Received signal strength indications (RSSI) could also be used to maintain network connectivity within a mesh network [4]. In [4], the mesh network is self-contained. For the applications of this paper, it is necessary to instead have the UAV network connected to a centrally located, powerful base station.

GAs have been used for route planning of UAVs, particularly in a system named DIANA (Dynamic Intelligent Autonomous Navigation Algorithm). DIANA uses a GA to plan a path for a UAV; however, DIANA does not coordinate the organization of multiple UAVs [8].

B. Genetic Algorithms

Once all of the parameters are known (coverage radius, height, and density map of network demand) a GA can be used to determine the best coverage map for the network users represented in the given density map. GAs are more efficient than guess-and-check methods because from each generation to the next, more is learned about the correct solution [9]. GAs consist of a series of steps: setup and a draw loop consisting of a fitness calculation, natural selection, mutation, and condition check [10].

The setup consists of filling a population with proposed solutions - a starting block of random guesses. The fitness function examines each proposed solution within the population and ranks it relative to an ideal solution. The draw function takes the value from the fitness function and performs natural selection to create a new generation [10]. To create a member of the next generation, two members of the previous generation are selected based on fitness value, and the information is combined to make a member of the new generation [11]. Mutation ensues regularly to ensure fixation on a particular solution does not occur [12]. This process is repeated until the population size is reached.

III. Proposed Algorithm

Efficiency is crucial for a temporary network of UAVs for search and rescuers to communicate, so both cost effectiveness and connectivity were maximized. We used tools including Friis Transmission Equation, geometry, trigonometry, a GA, as well as a new approach to information sharing to produce one cohesive algorithm, solving the problem of designing an optimal layout for UAVs to create a temporary network.

The input, provided by the user, to the proposed algorithm is a map of clusters of users defined as an ordered triple $(x, y, weight)$ where x and y represents the location of a grouping of users on the ground, and $weight$ is weight associated with that cluster of users - this can be the number of users or another priority level measure to determine the severity with which the location must be provided coverage. Parameters about the network connectivity of the UAV are also taken. As output, the algorithm returns a list of ordered triples representing the location of UAVs, (x, y, z), where this list covers the maximum number of users with the lease number of UAVs. This is a least-cost solution.

A. Height & Coverage Radius Calculations

A variant of Friis Transmission Equation [6] listed in equation (1) as well as the geometric and trigonometric equation denoted in equation (2) are used to perform the height and coverage calculations. The calculations performed are similar to that in [5]. The equations are expressed in a Python function and can be used with default data or data input by the user.

$$h = \frac{\lambda}{4\pi \times 10^{\frac{P_r - (P_t + D_t + D_r)}{20}}} \tag{1}$$

$$r = h \times tan(\theta) \tag{2}$$

The values for h and r are calculated using equations (1) and (2) and vary based on the input data.

Input : P, μ, $coverage\_radius$
Output: α

$\alpha \leftarrow []$
for $\rho \in P$ **do**
  $\chi \leftarrow []$
  for $UAV \in \rho$ **do**
    for $user\_cluster \in \mu$ **do**
      $d \leftarrow EuclideanDistance$(UAV,$\eta$)
      if $d \leq coverage\_radius \wedge user\_cluster \notin \chi$ **then**
        for $num\_users \in user\_cluster$ **do**
          $\alpha \leftarrow \rho$
        end
      end
    end
  end
end

Algorithm 1: Fitness Function

TABLE I: Glossary for Algorithms 1 & 2

Symbol	Description
α	The weighted population, with members of higher fitness added more times, and members with 0 fitness omitted
ρ	A proposed map, a member of the population
P	The population global variable
χ	The cluster exclusion list to avoid redundancies. A list of network user clusters that have already been provided coverage.
μ	The map density list - the list of all user clusters
N	The number of UAVs currently in use
ν	The mutation rate

Input : P, N, ν
Output: The new generation, stored in P

$P \leftarrow []$
$mutation\_check \leftarrow 0$
for $i \leftarrow 0$ *to* $|P|$ **do**
  $\rho \leftarrow []$
  for $m \leftarrow 0$ *to* N **do**
    r1, r2 $\leftarrow$ random integer [0, $|\alpha|$)
    if *mutation_check* = $\frac{1}{\nu}$ **then**
      change a value in the proposed map to a random integer [0, $|\alpha|$)
    else
    end
    $\rho \leftarrow$ (α[r1][m][0], α[r2][m][1])
  end
  mutation_check++
  $P \leftarrow \rho$
end

Algorithm 2: Draw Function

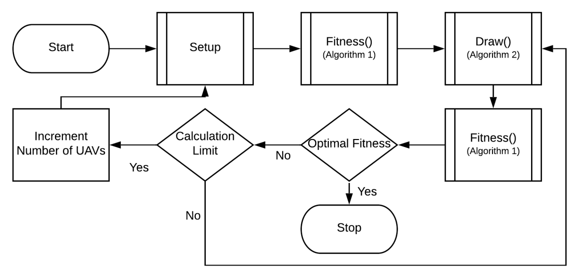

Fig. 2: Control loop - how fitness, draw, and iteration find an optimal solution

B. Genetic Algorithm

The GA is written as described in the background section: by conducting the necessary setup, and looping until a solution is found, using fitness, natural selection, and mutation. The setup fills the population with reasonable but random guesses as to what the answer could be. The following sections describe the specifications of the proposed algorithm.

1) Fitness Function: The fitness function ranks each proposed solution based on the bandwidth requirements the proposed network would cover. In addition to this, it ensures that no UAV has an unsupportable bandwidth demand. The maximum bandwidth supported per UAVs can be adjusted by way of a parameter, which is set to 300 Mbps by default because it is a common maximum bandwidth of WiFi routers. This number can depend on a series of factors, such as available bandwidth and signal strength. This is described further in Algorithm 1 and Table I.

2) Draw Function: Natural selection, mutation, and creation of the next generation occur in the draw function. Until the new population is full, information from two high-fitness solutions are combined, mutation occurs if appropriate, and the new offspring is added to the new population. This process is further described in Algorithm 2 and Table I.

3) Control Loop: The above GA, without the iterative aspect of the solution, cannot support multiple UAVs because it does not account for multiple variables. To solve this problem, an iterative approach is used. The algorithm runs with one UAV in the swarm, and if a solution is not found within a reasonable amount of time, the algorithm repeats with two UAVs. This process will repeat, adding an additional UAV every time, until a solution is found. “A reasonable amount of time” is defined as when the fitness of the model plateaus or, more technically, when a certain number of generations with the same best fitness score is reached. This number is defined by Equation 3:

$$limit = \left\lfloor \frac{N \times |\mu|}{3} \right\rfloor \tag{3}$$

where N is the number of UAVs currently being tested and $|\mu|$ is the number of clusters of users in the proposed problem.

Equation 3 was derived by concluding that the calculation limit should be a function of both the number of UAV in the solution and the number of user clusters, as they are the factors that determine the complexity of a proposed solution. The denominator of three was chosen by experimentation discussed further in Section IV-A-4. The process of using the iterative GA is illustrated in the flowchart in Fig. 2.

C. Information Inheritance

The algorithm as proposed is inefficient due to computational intensity and information wastefulness. If the calculation of an ideal coverage map is not possible with the given number of UAVs, the algorithm still computes the best layout for the UAVs available, by default. This fact is utilized by passing the information about optimal UAV layout forward in every iteration of the GA after adding a new UAV. The best possible layout is passed on by partially filling the population data structure. It cannot fill the entire population; however, because then the algorithm risks of losing variation. Equation 4 defines the number of members in the new population that will possess the optimal info found previously, and the verification for selection equation 4 can be found in Section IV-A-3.

$$inheritance = \left\lfloor \frac{|P|}{2} \right\rfloor \tag{4}$$

where $|P|$ is the population size for the GA.

The fact that the proposed algorithm uses data learned from past intermediate conclusions highlights this innovative approach compared to what has been done before. This concept, similar to transfer learning in that it shares and utilizes applicable information past what was originally intended [13], makes the proposed solution far more efficient than it is without the inheritance of information.

Before the inheritance can be added to the new population, another randomly placed UAV must be added to account for the necessary increase in the number of UAVs. The remainder of the population is to be filled with reasonable, random guesses as to an optimal solution.

D. Computational Complexity

The efficiency of the algorithm is derived to be:

$$O(G \times |P| \times N^2 \times |\mu|)$$

where G is the number of generations for which the proposed algorithm must run.

Given all of this information, the proposed system now produces a visualization similar to that depicted in Fig. 3. This display shows network user clusters as green dots representing a low density of network users, yellow dots representing a moderate density of network users, and red dots representing a high density of network users. The UAVs are depicted as blue dots with lighter blue circles surrounding that signify the provided coverage area. Such a diagram allows the user to more easily understand the numerical information outputted.

IV. Evaluation

Algorithm parameters, along with the fitness function and natural selection method were chosen based on experimentation to maximize performance. K-means clustering was used to compare efficiency and solution usability against the proposed algorithm.

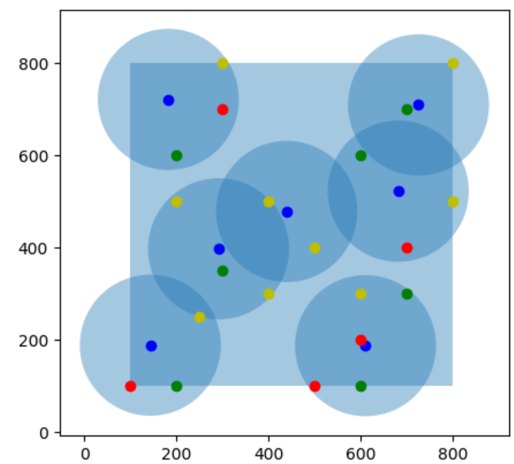

Fig. 3: Example of a map of network users to be covered by a temporary network with a possible solution of placement of UAVs to provide connectivity.

We assumed each UAV was equipped with a base station interface in addition to a backbone network using non-overlapping channels. The experiments were run on an Ubuntu machine with an Intel Core i5 processor, and the tests were run in such a way to only use one core of the quad-core processor as to achieve consistency between experimental runs.

A. Parameter Selection

There are four variables independent of the input data that affect the performance of the algorithm: population size, mutation rate, inheritance between GA iterations, and calculation limit.

Testing was done to find the best value for these parameters and provide the best performance by the algorithm. A map that presented above-average difficulties for the algorithm was used for this optimization. The population size was varied by increments of 100 between 300 and 700, and the mutation rate was varied between 0.005, 0.01, 0.02, 0.05, and 0.1. The inheritance and calculation limit are defined in equations (3) and (4), respectively, and the constants in the denominator of the equations were varied from one to four, and 0.5 was also tested.

For the four experiments, the following configurations were assumed: population size = 500, mutation rate = 0.01, inheritance between runs denominator coefficient = 2, and calculation limit denominator constant = 3, and the variables were manipulated thereafter.

The default parameter values were determined by previous informal experimentation. In each configuration, the experiment was run three times and the results were averaged. Two data points were collected from each trial: the number of UAVs used and the time elapsed while the algorithm was running. The number of UAVs was expected to be seven because the proposed algorithm, as well as k-means clustering,

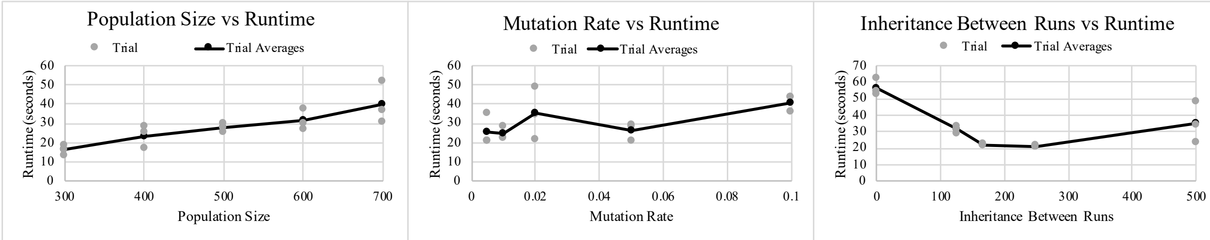

Fig. 4: Population Size, Mutation Rate, and Inheritance Experimental Data

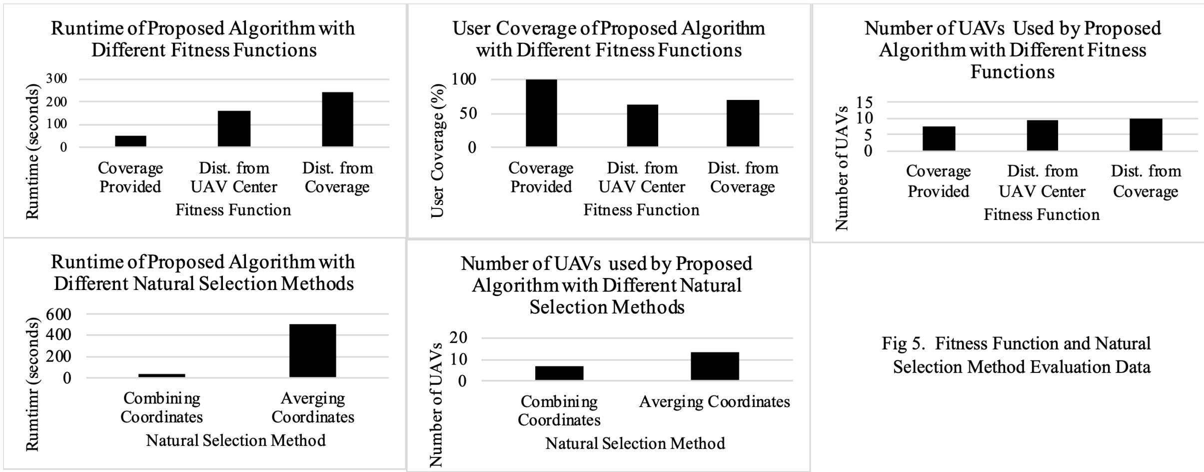

Fig 5. Fitness Function and Natural Selection Method Evaluation Data

never returned a value less than seven. Anything greater than seven would indicate that the algorithm did not find an optimal solution. Fig. 4 shows the results from these experiments.

1) Population Size Experimental Data: The population size of 400 was chosen from the data because it was the population size with the lowest average runtime, with the exception of 300, which did not yield an optimal solution in one of three trials.

2) Mutation Rate Experimental Data: The mutation rate of 0.01 was chosen from the data because it was the rate with the lowest average runtime, and it also found an optimal solution three out of three times.

3) Inheritance Experimental Data: The value with the least average runtime, 250, was chosen as optimal because testing returned an optimal solution for all three trials for two different inheritance values. With an experimental population size of 500, the optimal inheritance of 250 corresponds with equation (4), specifically having a denominator of two.

4) Calculation Limit Experimental Data: Equation (3) is a function of the number of UAVs and the number of network user clusters because these two factors determine the complexity of the solution and computation necessary to find the solution. As either or both of these variables increase, it is more difficult for the algorithm to compute a viable answer, so more time is allotted. Of particular interest is the constant in the denominator. Three was found to be the optimal solution because, though no value of the constant found an optimal solution three out of three times, the constant of three found an optimal solution two out of three times with the lowest average runtime within this class. Experimental data in Table II is provided as evidence.

TABLE II: Calculation Limit Experimental Data

Constant	Avg. Number of UAVs	Avg. Runtime (seconds)
4	7.67	22.833
3	7.33	26.853
2	7.33	40.189
1	7.67	86.334
0.5	7.67	167.65

B. Fitness Function & Natural Selection

To ensure the best possible performance, the proposed algorithm was executed with three different fitness functions and two different forms of natural selection. The best performing result was chosen in both cases.

The fitness functions quantify the validity of solutions. The tested fitness functions include the following: using the sum of the bandwidth provided to each user cluster, using the sum of the distances of all users from its closest UAV, and using the sum of the distances of all users from the coverage range of its closest UAV. All of the above fitness functions take into account weighting according to the density of the user cluster.

Using the bandwidth requirements as the fitness function was chosen over the other two options because it outperformed both. The average runtime and number of UAVs used was lower, while the bandwidth coverage was higher, all of which are the desired situations. The experimental data for the information previously listed is included in Fig. 5.

Natural selection is the process by which two parent proposed solutions create one child in the following generation.

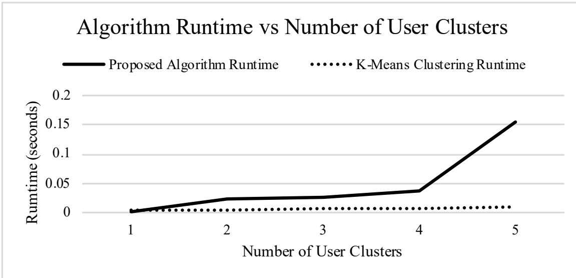

Fig. 6. Runtime Verification with K-Means Clustering

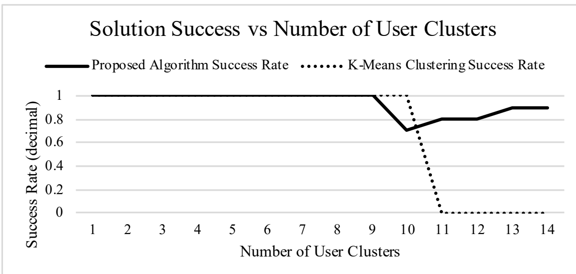

Fig. 7. Optimal Solution Verification with K-Means Clustering

The natural selection methods tested were: combining the x-coordinates from one parent and the y-coordinates from another and averaging the x-coordinates and y-coordinates from each parent. The experimental data for the information previously listed is included in Fig. 5.

Combining an x-coordinate from one parent and the y-coordinate of another was used rather than averaging because it outperformed the averaging method in every category.

C. K-Means Clustering Comparison

We verified the proposed algorithm by comparing the performance of the proposed algorithm with the results of k-means clustering. The experimentation was conducted by varying the number of clusters of network users between one and 14, and run each for 10 trials. Each of these configurations was run with both the proposed system and k-means clustering. Runtime and solution validity were recorded, compared, and charted. A random number generator was used to create locations of network user clusters on the map.

The performance of the proposed algorithm was compared to that of k-means clustering algorithm in two ways: runtime and success in finding a solution.

The experimental data are included in Fig. 6 and Fig. 7. Observations from analysis of this data are: The iterative GA does not execute as quickly as k-means clustering. The proposed algorithm and k-means clustering find solutions equally well with simple problems. As problem complexity increases, the system finds an optimal solution more consistently.

V. Conclusions and Future Work

We presented an algorithm to find an optimal layout for UAVs in a swarm, given a density map of network users attempting to search an area. The proposed system uses calculations, a GA, and information transfer to provide an efficient way to generate an optimal solution.

The proposed algorithm was evaluated against variations of itself and k-means clustering algorithm to ensure that it presents ideal solutions. The results of the evaluation yielded that, while the proposed algorithm runs more slowly, it also more consistently yields an optimal solution when presented with a complex map of users.

We aim to expand upon this work by allowing re-positioning of UAVs, given a new layout of network users, and using other machine learning architectures. An algorithm that can support a 3-dimensional version of the problem for varying UAV heights could be attempted.A graphical user interface should be added for better ease-of-use.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. IIS-1757929. 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 National Science Foundation.

References

[1] M. Alzenad, A. El-Keyi, F. Lagum and H. Yanikomeroglu, "3-D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage," in IEEE Wireless Communications Letters, vol. 6, no. 4, pp. 434-437, Aug. 2017.

[2] E. Kalantari, H. Yanikomeroglu and A. Yongacoglu, "On the Number and 3D Placement of Drone Base Stations in Wireless Cellular Networks," 2016 IEEE 84th Vehicular Technology Conference (VTC-Fall), Montreal, QC, 2016, pp. 1-6.

[3] A. N. Patra and S. Sengupta, "Dynamic deployment of UAV-enabled floating access points for serving hot zones," 2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), Seattle, WA, 2017, pp. 1-8.

[4] Tristan Brodeur, Paulo Regis, David Feil-Seifer, and Shamik Sengupta. "Search and Rescue Operations with Mesh Networked Robots." In IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (IEEE UEMCON), Nov 2018.

[5] P. A. Regis, A. N. Patra and S. Sengupta, "Unmanned Aerial Vehicles Positioning Scheme for First-Responders in a Dynamic Area of Interest," 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 2018, pp. 1-5.

[6] H. T. Friis, "A Note on a Simple Transmission Formula," in Proceedings of the IRE, vol. 34, no. 5, pp. 254-256, May 1946.

[7] W. Shi, H. Zhou, J. Li, W. Xu, N. Zhang and X. Shen, "Drone Assisted Vehicular Networks: Architecture, Challenges and Opportunities," in IEEE Network, vol. 32, no. 3, pp. 130-137, May/June 2018.

[8] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-6.

[9] A. Majd, A. Ashraf, E. Troubitsyna and M. Daneshtalab, "Integrating Learning, Optimization, and Prediction for Efficient Navigation of Swarms of Drones," 2018 26th Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP), Cambridge, 2018, pp. 101-108.

[10] Whitley, D. Stat Comput (1994) 4: 65.

[11] Kadri, R. and Boctor, F. (2018). An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case. European Journal of Operational Research, 265(2), pp.454-462.

[12] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," in IEEE Transactions on Systems, Man, and Cybernetics, vol. 24, no. 4, pp. 656-667, April 1994.

[13] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," in IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345-1359, Oct. 2010.