Metaheuristic-based optimal 3D positioning of UAVs forming aerial mesh network to provide emergency communication services

Manali Gupta | Shirshu Varma

Abstract
The optimal placement of unmanned aerial vehicles (UAVs) to facilitate post-disaster emergency communication services is a paramount research domain. The novelty of the authors’ work is to optimally place available UAVs in 3D space to meet the objectives prominent during such situations. The objectives considered here are target coverage, QoS, energy consumption and two newly characterized objectives, i.e. equal load distribution over UAVs and fault tolerance for improving network connectivity and lifetime. To improve these conflicting objectives altogether, the authors proposed two metaheuristic-based hybrid optimization algorithms, namely, HWWO-HSA, a hybrid of Water Wave Optimization (WWO), Harmony Search (HS) and Simulated Annealing (SA); and HGA-SA, a hybrid of Genetic Algorithm (GA) and SA. Furthermore, to improve network performance, the authors performed their parameter tuning using well-known Taguchi’s design of experiment. For maintaining network connectivity, the authors used the concept of graphs for connected components. The proposed hybrids are compared with their originals, i.e. GA, HS, SA and WWO over different scaled test scenarios. The non-parametric Kruskal–Wallis test and Dunn’s post hoc test suggest that hybrids perform better than the originals. HGA-SA significantly outperforms in small-scale scenarios while HWWO-HSA significantly outperforms in medium-scale and large-scale scenarios. However, being hybrid the computational time of the hybrids is greater than the originals.

1 | INTRODUCTION

The easy deployment and cost-effective feature of unmanned aerial vehicles (UAVs) compared to human-piloted planes, balloons and aerostats have made UAVs quite popular in the field of information technology. The various application areas of UAVs are search and rescue [1], monitoring [2], detection [3], communication facilities in a disaster-struck region [4], integration with 5G networks [5], etc. The network formed using these UAVs further enhances their capability and usability by integrating features of rich interconnection and interaction required to perform highly skillful and intellectual tasks. Among such networks, Aerial Mesh Network (AMN) is quite famous. These networks connect UAVs to each other wirelessly in the sky and arrange them in mesh topology [6]. The mesh topology makes the network robust and fault tolerant and ability to fly makes the network easily deployable and configurable in difficult situations.

One such difficult situation could be the one where existing infrastructures like base stations or towers, that provide communication facilities in a region, have been damaged or destroyed during a disaster. To facilitate emergency communication services during cellular blackout in such a disaster-struck area, AMNs can prove to be a good alternative [4, 7]. Figure 1 describes such a scenario where an AMN is deployed that acts like a relay network to facilitate communication between the natives of that region and the control stations. The control station will then take necessary actions in accordance with the situation to assist the people.

Apart from its various uses, there are certain network issues which need to be centred for enhancing the performance of an AMN. These network issues can be target coverage, quality of
service (QoS), energy consumption, connectivity, deployment cost (or number of UAVs), network lifetime, fault tolerance and robustness, security, etc. However, it has been observed that most of these issues are greatly affected by the positioning of UAVs [8–10]. Furthermore, the optimal positioning of UAVs to improve these issues is mostly observed as NP-hard problems [9–11]. For such problems, well-known metaheuristic-based optimization algorithms like genetic algorithms (GA), swarm-based algorithms, bio-inspired algorithms, etc. have proved to be a good solution. The NP-hard problems whose exact optimal solutions do not exist, metaheuristic-based optimization algorithms give near to optimal (or sub-optimal) solutions to such problems within a reasonable time. These optimization algorithms can search for solutions within a large continuous search space where UAVs can be positioned at any point in 3D space of the given target area. Also, these algorithms require less mathematical modelling for the behaviour of each and every UAV of the network, in contrast to the traditional gradient-based techniques. Please note that finding an exact optimal solution for an NP-hard problem is not possible, therefore, we called sub-optimal solutions as optimal solutions in the rest of the paper to keep the things simple.

On this concept, Sabino et al. [8] aimed at facilitating differentiated data requirements for the end users with minimum deployment cost through optimal placement of UAVs using multi-objective evolutionary algorithms. Caillouet and Razafindralambo [9] centred on objectives, i.e. QoS, full target coverage and minimum deployment cost along with maintaining network connectivity through optimal deployment of UAVs using ε-constraint method. Reina et al. [10] used a multi sub-population-based GA for optimal deployment of UAVs in order to meet target coverage, fault tolerance and data redundancy. To deal with fault tolerance, Ref. [10] considered to maximize the size of minimum vertex cuts, whose exclusion can disconnect the whole network. In addition to this, we contributed to improve fault tolerance of the network by minimizing the number of such minimum vertex cuts and maximizing the number of UAVs connected to the control station. This keeps the network to remain connected for longer time. Furthermore, we focused on equal distribution of load over UAVs so that the UAVs get less unequally drained out of energy, and thus, keep the network connected for longer time. In summary, while considering the familiar network issues, i.e. target coverage, QoS and energy consumption of the network, we also focused on equal load distribution and fault tolerance of the network that are important when providing emergency communications in post-disaster regions.

Thus, the novelty of our work is to optimally place the available number of UAVs in 3D space to improve the above-mentioned network issues/objectives while keeping the network connected. In one of our prior work [12], we have considered objectives like target coverage, QoS and energy consumption by optimizing the altitude of UAVs arranged in 2D fixed grid structure using some optimization algorithms. The reason to consider the grid structure is to deal with the issue of connectivity. This paper used the concepts of graph theory for connected components in place of the grid structure to address the issue of connectivity. The concept of connected components helps to optimize the positions of UAVs in 3D space, which adds more flexibility to the system in comparison to the system proposed in [12], that only optimizes the altitude of the grid formed by UAVs having same altitude. Furthermore, this paper proposed two novel network objectives, i.e. equal load distribution and fault tolerance. The equal load distribution helps to maintain a balance of resource utilization between homogenous UAVs. It assures that each UAV should cover an equal number of targets so that the load gets equally distributed between all homogenous UAVs. This improves network connectivity as the number of UAVs died due to unequal resource drainage will decrease, which decreases the probability of network for getting disconnected due to dead UAVs, and thus, improving the overall network lifetime. Another network objective, i.e. fault tolerance keeps a check on the weaker connections present in the network in order to maintain the network connectivity for longer time. To meet the above-mentioned network objectives altogether, we proposed new hybrid optimization algorithms. We have further improved the network’s performance by tuning the parameters of these hybrid optimization algorithms using Taguchi’s Design of Experiment (DoE) [13], which is not incorporated in many previous works in the past.

The target coverage objective focuses on improving the number of targets covered by the network. The QoS objective focuses on improving QoS provided by the network while the energy consumption objective focuses on reducing the total energy consumed by the network during UAVs’ flight and data transmission. However, the objectives, i.e. target coverage, QoS and energy consumption get inversely affected with respect to the altitude of UAVs [9, 14]. The equal load distribution focuses on the objective that each UAV should cover an equal number of targets, which may reduce the target coverage when the
targets are densely distributed. The reason is when the targets are densely distributed then a UAV cannot cover more than a certain number of targets to keep the equal load distribution, and thus, its coverage decreases. Furthermore, fault tolerance keeps the UAVs closer when the targets are sparsely distributed in order to keep the network connected, and thus, may reduce both target coverage and equal load distributions. As we can see that these objectives, which have different intended purposes, may have an inverse effect on each other while positioning UAVs. Therefore, optimizing the placement of UAVs in a given 3D space, which is a large continuous search space, is quite challenging. Furthermore, most of the optimal placement problem of UAVs is observed as NP-hard [9–11]. To trade-off these conflicting objectives simultaneously, we used a metaheuristic-based optimization approach to get the optimum positions of UAVs. The metaheuristic-based optimization algorithms apply mathematical-based search models over data collected while network simulation to find the optimal solutions.

The optimization algorithms use both exploitation (i.e. local search) and exploration (i.e. global search) to find the optimal results of a particular problem. However, it is not possible for any optimization algorithm to solve each and every optimization problem through proper balancing of both exploitation and exploration (i.e. No Free Lunch [NFL] theorem). In the past, various hybrid optimization algorithms are designed to solve different optimization problems [15–17]. This inspired us to develop new hybrid optimization algorithms which properly balance exploitation and exploration search ability to solve our optimal placement problem of UAVs. This study presents the first attempt to hybridize Water Wave Optimization (WWO) with Harmony Search (HS) and Simulated Annealing (SA). The strong exploration capability of WWO in both small and large search spaces is integrated with the exploitation capability of HS and Metropolis Monte Carlo criteria of SA for removing random unfit solutions at later execution phase of the hybrid, which causes its faster convergence. Moreover, we proposed a hybrid of GA and SA that balances both exploration and exploitation and has lesser computational time (CT) in comparison to the prior hybrids of GA and SA [18–20]. By performing experiments, it is observed that the proposed hybrid optimization algorithms perform quite efficiently than their original ones, i.e. GA, HS, SA and WWO while finding optimal locations of UAVs in 3D space.

On the basis of the proposed research work, a brief literature review is presented in Section 2. Section 3 elaborates the system model of our work. Section 4 describes the proposed hybrid optimization algorithms for optimal positioning of UAVs in our system model. Furthermore, Section 5 designs the optimal solution framework for our system model. Section 6 presents the results of each tuned optimization algorithm and compared them in different test scenarios to find the best one in each scenario. Furthermore, Kruskal–Wallis test (KW test) and Dunn's post-hoc test are used to validate the results. Section 7 presents the software and hardware challenges in our system model. Section 8 concludes the whole research work of the paper with future scopes.

2 LITERATURE REVIEW

The problem of optimized positioning of UAVs to perform a particular task mostly covers under NP-hard problems [21]. This is because searching an optimal solution in huge multi-dimensional and multi-modal solution space is not feasible in polynomial time. The optimized positioning of UAVs could be optimizing UAV altitude at fixed position [21] or UAV position at fixed altitude [22] or both [9]. The positioning of UAVs directly affects its coverage area [10]. Furthermore, coverage has been categorized as static and dynamic based on the movement of UAVs. If UAVs hover over a particular point during their flight time, it is called static coverage while if they keep moving from one point to another, it is called dynamic coverage [23]. In this work, we have preferred static coverage due to the interest of convenience.

Apart from coverage, there are other issues which get affected by the positioning of UAVs. The issues include service quality, deployment cost, energy consumption, connectivity, mission completion time, network lifetime, interference, etc. These issues could be considered either as objectives or constraints based on the system requirements which cause change in the definition of the problem as well as its respective solution. Mozaffari et al. [14] centred over target coverage and network lifetime by applying circle packing theory to optimally place UAVs in 3D space. Reina et al. [10] aimed to meet target coverage, fault tolerance and data redundancies by applying multi sub-population-based GA for optimal positioning of UAVs. Furthermore, Caiillouet et al. [9] centred on QoS and full target coverage while maintaining network connectivity, by optimally deploying minimum number of UAVs using epsilon-constraint method. Hu et al. [24] incorporated smallest enclosing circle concept with evolutionary algorithm for UAV's optimal placement to ensure reliable communication in target area using minimum number of UAVs. However, there could be a condition when we cannot accommodate minimum number of UAVs to meet the defined goals. Furthermore, there is a need to focus on equal load distribution and the weak connections present in the network to keep the network connected for longer time. Apart from familiar network objectives like target coverage, energy consumption, QoS, this study also focused on equal load distribution and fault tolerance to improve network performance in critical situations.

The metaheuristic-based optimization algorithms, capable of solving any complex optimization problem, search the solution space using both exploitation and exploration property. The exploitation property makes the algorithm to rigorously search the local solution space while the exploration property makes the algorithm to fully explore the global space. However, according to No Free Lunch (NFL) theorem, it is not possible for any optimization problem to solve each and every optimization problem. In the past, various hybrid optimization algorithms are developed to solve different optimization problems. Zulj et al. [15] designed a hybrid of adaptive large neighbourhood search (ALNS) and tabu search (TS) to solve the order-batching problem which is observed as NP-hard problem. Senel et al. [16] proposed a hybrid of particle swarm optimization
(PSO) and grey wolf optimizer (GWO) to solve one of the toughest industrial problem, i.e. leather nesting. Murthy et al. [17] developed a hybrid of PSO, cuckoo and chemical reaction optimization algorithms to solve the problem of hiding association rules in database transactions.

On the basis of NFL, we proposed HWWO-HSA which is a hybrid of WWO, HS and SA. The WWO algorithm has strong exploration but weak exploitation property due to its random function used in its propagation phase. Earlier, Singh et al. [25] developed a hybrid of WWO and sequential quadratic programming (SQP) where SQP is used for exploitation while WWO for exploration. Furthermore, Sahebjamnia et al. [26] developed hybrid of WWO and GA where the crossover operation of GA is used in the propagation phase of WWO for its better exploitation. Rong et al. [27] designed a hybrid of WWO and PSO where PSO uses its property for faster convergence and exploitation along with exploration property of WWO. This study proposed HWWO-HSA which used the strong exploitation property of HS and Metropolis Monte Carlo criteria of SA, for discarding random unfit solutions at the later execution phase of the hybrid, along with the exploitation property of WWO. It is observed through experiments that the new hybrid balances between both exploitation and exploration with faster convergence to find optimal locations of UAVs.

Also, we proposed a hybrid of GA and SA which improves the exploration ability of GA using SA’s random search for solutions in large search spaces. Earlier, Mamaghani et al. [18], Huang et al. [19], Anto et al. [20], etc. have designed hybrid of GA and SA. However, these earlier designs have high time complexity which is not suitable for our problem. The proposed HGA-SA is designed in such a way that it has comparatively much lesser time complexity as compared to earlier ones.

3 SYSTEM MODEL

Here, a system model is described to provide a better insight to the situation where there is need to optimize the performance of AMNs. In this model, we have $M$ number of targets that are present in a disaster struck area $A$. The base stations that used to provide communication facility in such an area are collapsed. Therefore, to facilitate communication in the given area, we deploy an AMN using UAVs. The wireless communication between UAVs and targets is supported by IEEE 802.11 g (Wi-Fi) technology.

We assume $N$ available number of UAVs, forming an AMN, to be deployed in the given area $A$. The UAVs should be placed in 3D space in such a way that they remain connected with each other and with the control station. Therefore, consider the control station while finding whether the network is connected or not. To check network connectivity, we used the concept of graph theory dealing with connected components in undirected graphs. The reason to consider undirected graphs is that the transmission range of each UAV is assumed to be equal. In this concept, we count the number of connected components in a graph using both depth first search and breadth first search from any vertex. If the number of connected components is one, then the graph is connected otherwise not. Some examples of connected components in graphs are shown in figure 2.

4 METHODOLOGIES USED

In order to deal with optimal placement problem of available number of UAVs for increasing coverage of given targets, QoS and equal load distribution with reduced energy consumption by the network, we use popular metaheuristic optimization approaches. This is because these kinds of problems are usually multi-modal and multi-dimensional and mostly come under NP-hard problems, as discussed earlier, which has no exact solution and usually converge near to optimal (or sub-optimal solutions) [28]. To keep things simple sub-optimal and optimal solutions are considered same in all over the paper. The metaheuristic-based optimization approach uses mathematical models over collected data to find optimum solutions. In this section, we presented proposed metaheuristic-based hybrid optimization algorithms, i.e. HWWO-HSA and HGA-SA along with original ones, i.e. GA [29], HS [30] and SA [31], WWO [32]. Each of these algorithms is discussed one by one.

4.1 Genetic algorithm

GA [29] is influenced by nature’s behaviour of evolution. It evolves the generations using biological operator such as selection, crossover and mutation. These operators are applied over each individual/chromosome of the population of size $P$. At each generation, two parents are selected based on their fitness values using selection operator like Roulette wheel, tournament selection, etc. These parents perform crossover operation with probability $P_c$ to produce two offspring. Each offspring further undergo through mutation with probability $P_m$ to get out from
local maxima/minima. This algorithm will stop either at convergence or maximum number of iterations $MaxIt$.

According to our system model, we assume a given area where we deploy $N$ available UAVs in 3D space. Now create population $Pop$ of size $P$ which contains connected AMNs. To find whether the network is connected or not, the concept of connected components is used (Section 3). Each of these connected AMNs act as a chromosome with their corresponding fitness values. These chromosomes undergo through selection, crossover and mutation operations. If an individual (or AMN) generated after such operations is not a connected network, discard it and again perform the operations. After maximum number of iterations, we find the optimal chromosome or connected AMN with best fitness value.

**Individual structure:**

An individual (i.e. either chromosome or particle), used by an optimization algorithm, is defined as an AMN formed using $N$ UAVs that are placed in 3D space. Therefore, an individual, denoted as $p$, is structured as:

$$p = \{(x_1, y_1, h_1), (x_2, y_2, h_2), \ldots (x_N, y_N, h_N)\},$$

where $(x_i, y_i, h_i)$ is the position of $i$th UAV in 3D space. Furthermore, there is a need to be assured that each individual remains connected throughout the execution of an algorithm. To check whether an individual is connected or not, find out its number of connected components. If the number of connected components is one, only then the network is said to be connected otherwise not (Section 3). If an individual is not connected, then discard it and generate new one.

### Pseudo code of GA (Maximization problem)

1. For a given area $A$, place available $N$ UAVs in 3D space with their heights within minimum and maximum values. These UAVs will form an AMN.
2. Set transmission range of each UAV as $R$ which makes the network to be treated as an undirected graph.
3. Set parameters $Pc$, $Pm$, $P$, $MaxIt$
4. Input: Initialize population $Pop$ of size $P$ which contains randomly created connected AMNs represented as chromosomes. To check whether network is connected or not, count its number of components. If the count is one, then it is connected otherwise not.
5. Calculate fitness value $f(p)$ of each chromosome $p$.
6. for iteration $= 1$ to $MaxIt$ do
7. Set $Pop_{new} \leftarrow \emptyset$
8. for $i = 1$ to $P$ do
9. $x, y \leftarrow$ Select_random($\{1..Pop\}$, $Pop$)
10. $z \leftarrow$ Crossover($x, y, P$)
11. $z \leftarrow$ Mutate($z, Pm$)
12. $Pop_{new} \leftarrow Pop_{new} \cup z$
13. endfor
14. $Pop \leftarrow Pop_{new}$
15. endfor
16. $Best_Solution \leftarrow \text{best}(Pop)$

### Pseudo code of HS (Maximization problem)

1. Set parameters $HMCR, PAR, P, MaxIt$
2. Input: Initialize memory $HSM$ with random $P$ solutions
3. for iteration $= 1$ to $MaxIt$ do
4. for $j = 1$ to $d$ do
5. if rand(0, 1) $<$ $HMCR$
6. $x_{new} \leftarrow$ Select_random($Pop$)
7. $x_{new} \leftarrow$ Mutate($x_{new}$, $PAR$)
8. else
9. $x_{new} \leftarrow$ $LB +$ rand(0,1)($UB - LB$)
10. endif
11. endfor
12. if $f(x_{new}) > f($Best($HSM$))
13. $HSM \leftarrow HSM -$Best($HSM$)
14. $HSM \leftarrow HSM \cup x_{new}$
15. endif
16. endfor
17. Best_Solution $\leftarrow$ Best($HSM$)

### 4.2 Harmony search

The HS optimization [30] uses the principle used by a musician to compose a new harmony. The musician tries different music pitches stored in his/her mind to compose a new harmony. This analogy is used by HS to solve an optimization problem using all solutions stored in a memory to get the optimal solution. For a $d$-dimensional problem, HS initializes a memory, called as HS memory or HSM, of size $P$ with random solutions where each solution has $d$ dimensions/components. At each iteration, it either randomly selects any solution in HSM using Harmony Memory Considering Rate (HMCR) or generates a random value lying within lower bound (LB) and upper bound (UP) with probability (1-HMCR). If it selects a solution from HSM then assign its $j$th component to $j$th component of a new solution and further mutate it using Pitch Adjusting Rate (PAR). If the generated new solution is better than the worst solution in HSM, then replace the worst solution with the new solution, otherwise, discard it. Thus, we can say that HS has both crossover and mutation operation, similar to GA. However, HS uses all members in HS memory to generate a new solution while GA uses only two solutions for crossover. After maximum iterations of HS, the optimal solution is generated.

### 4.3 Simulated annealing

SA [31] is based on the relationship between temperature and motion of particles. If the temperature is high the randomness in motion is high and if it is low the randomness in motion is low. Therefore, at initial stage we search for all possible positions with high randomness and at later stage we try to converge
to the optimal solutions while keeping less randomness to select the positions using Metropolis Monte Carlo criteria. In order to have lower temperature at later stages, we decrease the temperature at the rate of alpha, $\alpha$, where $0 \leq \alpha \leq 1$, after each iteration. However, each temperature update is maintained for a number of sub-iterations. Stable. After maximum number of iterations, the optimal solution is obtained.

### 4.4 Water wave optimization

The WWO [32] is based on the concepts of water waves that lie in different regions of the sea. The waves lying in shallow water are more still in comparison to the waves lying in deep water. The purpose of WWO is to make those waves that lie in deep water (having low fitness) to come up in the shallow water (having high fitness). It initializes a population $Pop$ of $P$ random waves having height $h_{\text{max}}$ and their fitness value is inversely proportional to their height from the water surface. At each iteration, a wave undergoes propagation by randomly choosing a position in each dimension. Also, the wavelength of the wave is updated using wavelength reduction coefficient $\mu$. If this new position is better than current one, then update it with new one. Furthermore, if it is the new best position till now then break it up to $k_{\text{max}}$ number of solitary waves. Select the best solitary wave and if it the best position till now then make it the new best position. Then, break operation is performed with breaking coefficient $\beta$, to search around the new best position for finding the best one in that region. However, if the new position generated through propagation is not better than current one, then decrease its wave height by one. We perform refraction of the wave when its height becomes zero that prevents search operation from getting stuck into local optima. After maximum iterations, the algorithm produces an optimal solution.

#### Pseudo code of WWO (Maximization problem)

1. Set $h_{\text{max}}, k_{\text{max}}, \beta, \mu, P, \text{MaxIt}$
2. **Input**: Initialize randomly population $Pop$ of $P$ waves having height $h_{\text{max}}$
3. $x^* \leftarrow \text{Best}(Pop)$
4. **for** $i$ = 1 to $\text{MaxIt}$ **do**
5.   **for** each wave $x$ in $Pop$ **do**
6.     $x \leftarrow \text{Propagate}(x_{\text{new}})$
7.     $\lambda \leftarrow \text{Update}(\lambda, \mu)$ /* $\lambda$ is wavelength*/
8.     **if** $f(x_{\text{new}}) > f(x)$ **then**
9.       **if** $f(x_{\text{new}}) > f(x^*)$ **then**
10.          $x^* \leftarrow \text{Best}(\text{Break}(x_{\text{new}}, \beta, k_{\text{max}}))$
11.       **if** $f(x) > f(x_{\text{new}})$ **then**
12.          $x_{\text{new}} \leftarrow x$
13.       **endif**
14.          $x^* \leftarrow x_{\text{new}}$
15.       **endif**
16.     **endfor**
17. **endfor**

### 4.5 Hybrid GA-SA

GA uses biological operators to search the solution globally yet may suffer from premature convergence due to mutation operator which becomes incapable to explore solutions in large search spaces. On the other hand, SA randomly searches for global solution in large search spaces with lesser chances of premature convergence due to the probability of accepting new solution even when it is not better than previous one using Metropolis Monte Carlo criteria. However, SA have lower convergence rate. The proposed hybrid of GA and SA (i.e. HGA-SA) combines good properties of both GA and SA to give global optimal solution with higher convergence rate. Furthermore, HGA-SA is designed in such a way that it has comparatively much lesser time complexity as compared to earlier ones in [18–20]. First, it applies GA operators like parent selection, crossover and mutation to get the new population. If the new population’s best chromosome has not improved from the previous one, SA is applied to find the better solution. If SA finds better solution, it will be merged to the new population while discarding the chromosome having least fitness value; otherwise the new population remains unchanged. After maximum number of iterations, the optimal solution will be produced.

#### Pseudo code of HGA-SA (Maximization problem)

1. Set parameters $Pc, Pm, T, \alpha, P, \text{MaxIt}$
2. **Input**: Initialize population $Pop$ of size $P$ which contains randomly created connected AMNs represented as chromosomes
3. Calculate fitness value $f(p)$ of each chromosome $p$
4. **for** $i$ = 1 to $\text{MaxIt}$ **do**
5. /*Run GA*/
6. **Set** $Pop_{\text{new}} \leftarrow \varnothing$
7. **for** $i$ = 1 to $P$ **do**
8. Perform GA operations, i.e. parent selection, crossover and mutation and generate $Pop_{\text{new}}$
9. **endfor**
10. **if** $f(\text{best}(Pop_{\text{new}})) \leq f(\text{best}(Pop))$ **then**
11. $p \leftarrow \text{best}(Pop_{\text{new}})$
12. **while** $k = 1$ **do**
4.6 | Hybrid Water Wave optimization with Harmony Search and Simulated Annealing

The hybrid of WWO, HS and SA applies useful features of HS and SA over WWO. The random function of WWO, used for propagation, cannot fully exploit the search space over a given region and may get out of that region after some time using its quick exploration capability. To improve the exploitation and faster convergence of WWO, we incorporated the concept of harmonics of HS. The HS makes use of every member of HS memory to generate a component of new solution with probability of HMCR and randomly generate a component of a solution with probability of 1-HMCR. This exploitation property of HS replaces the random propagation of WWO to improve its exploitation property. Furthermore, we observed that during exploration phase of WWO, its refraction operation accepts new solution even when the new one is not better than the current one. Here, we used the Metropolis Monte Carlo criteria of SA to accept the new unfit solution with certain probability. This probability is higher during initial phase when the temperature is high in order to have more exploration of the search space. At later phase, the probability of exploration decreases with decrease in temperature in order to improve convergence to the global optimum. After maximum iterations, the optimal solution is achieved.

Pseudo code of HWWO-HSA (Maximization problem)

1. Set parameters HMCR, PAR, T, α, h_max, k_max, β, P, MaxIt
2. Input: Initialize randomly Pop of P waves having height h_max

/*Run SA*/
12. Select a new point q from neighbourhood of p
13. \[ \Delta \leftarrow f(q) - f(p) \]
14. if \( \Delta > 0 \) or \( \exp(\Delta/T) \geq \text{rand}(0,1) \)
15. \( p \leftarrow q \)
16. else
17. \( p \) is unchanged
18. \( k \leftarrow 0 \)
19. endif
20. endwhile
21. \( T \leftarrow \alpha \times T \)
22. endif
23. if \( f(q) \neq f(\text{best}(\text{Pop}_{new})) \)
24. \( \text{Pop}_{new} \leftarrow \text{Pop}_{new} \cup q \)
25. \( \text{Pop}_{new} \leftarrow \text{worst}(\text{Pop}_{new}) \)
26. endif
27. \( \text{Pop} \leftarrow \text{Pop}_{new} \)
28. endfor
29. \( \text{Best Solution} \leftarrow \text{Best}(\text{Pop}) \)
30. \( x^* \leftarrow \text{Best}(\text{Pop}) \)
31. for iteration = 1 to MaxIt do
32. for each wave \( x \) in \( \text{Pop} \) do
33. for each component \( j = 1 \) to \( d \) /*d is dimension*/
34. /*Start HS exploitation*/
35. if \( \text{rand}(0,1) < \text{HMCR} \)
36. \( x'_{\text{new}} \leftarrow \text{Select_random}(\text{Pop}_p) \)
37. \( x'_{\text{new}} \leftarrow \text{Mutate}(x'_{\text{new}}, PAR) \)
38. else
39. \( x'_{\text{new}} \leftarrow \text{LB} + \text{rand}(0,1) \times (\text{UB} - \text{LB}) \)
40. endif
41. endfor
42. /*End HS exploitation*/
43. if \( f(x'_{\text{new}}) > f(x^*) \)
44. \( x^* \leftarrow \text{Best}(\text{Break}(x'_{\text{new}}, \beta, h_{\text{max}})) \)
45. if \( f(x') > f(x^*) \)
46. \( x'_{\text{new}} \leftarrow x' \)
47. endif
48. \( x^* \leftarrow x'_{\text{new}} \)
49. endif
50. \( x \leftarrow x'_{\text{new}} \)
51. else
52. \( x \times h \leftarrow \text{Xh} - 1 \)
53. if \( x \times h = 0 \)
54. \( x_{\text{new}} \leftarrow \text{Refract}(x) \)
55. if \( f(x_{\text{new}}) > f(x^*) \)
56. \( x \leftarrow x_{\text{new}} \)
57. endif
58. endif
59. endif
60. endfor
61. endfor
62. endfor
63. \( \text{Best Solution} \leftarrow x^* \)

5 | OPTIMIZATION FRAMEWORK

The UAVs’ optimum positioning in 3D space optimizes the following network objectives, i.e. improved target coverage, QoS, equal load distribution and fault tolerance with reduced energy consumption. The optimization framework, as shown in
Figure 3, uses metaheuristic-based optimization algorithms, one at a time, to produce optimal position of each UAV in 3D space.

An optimization algorithm generates a solution or a set of solutions at each iteration, where each solution represents a connected AMN defined by the position of UAVs. To evaluate the objectives of an AMN like target coverage, QoS, energy consumed, equal load distribution and fault tolerance, an AMN is simulated using the network simulator (ns-2). The simulator evaluates values of these objectives, which further summed up to get the fitness value of each solution (Equation 17). Based on these fitness values, the optimization algorithm uses its optimization technique to select a better solution among all solutions. After running maximum iterations, the algorithm produces an optimal solution/AMN having optimal positioning of UAVs as shown in Figure 3.

At a given disaster-struck area $A$ having randomly scattered set of targets/end devices $E$ (discussed in Section 3), a set of UAVs $U$ is positioned in 3D space to facilitate communication services. Furthermore, height of a UAV $u \in U$ should be bound to a minimum and maximum value, defined by users, as follows:

$$h_{\text{min}} \leq h_u \leq h_{\text{max}}.$$  \hspace{1cm} (1)

Now, we define each objective function one by one.

### 5.1 Target coverage

In order to define $TC$, we first define the coverage radius $r_u$ of a UAV $u$ as follows [14]:

$$r_u \leq h_u \cdot \tan (\Theta / 2),$$  \hspace{1cm} (2)

where $\Theta$ is UAV’s visibility angle. Now, target $e \in E$ located at $(x_e, y_e, 0)$ is within coverage radius of UAV $u$ located at $(x_u, y_u, h)$ if the distance between target $e$ and projection of UAV $u$, i.e. $s(e, u)$ is less than the coverage radius $r_u$, i.e.

$$s(e, u) \leq r_u, \hspace{1cm} \text{(3)}$$

where $s(e, u) = \sqrt{(x_e - x_u)^2 + (y_e - y_u)^2}$.

Now, the objective, i.e. $TC$, which is defined as the total number of targets covered by the network [14], is formulated as:

$$TC = \sum e_u, \hspace{1cm} \text{(4)}$$

where

$$e_u = \begin{cases} 1 & \text{if target } e \text{ is closest to any UAV } u \in U \text{ and } s(e, u) \leq r_u \\ 0 & \text{else.} \end{cases}$$

The variable $e_u$ is set to 1 only if target $e$ is closest to any UAV $u \in U$, i.e. the distance between target $e$ and projection of UAV $u$ at the ground is smallest as compared to the distance between target $e$ and projection of other UAVs at the ground, and target $e$ is within coverage radius $r_u$ of UAV $u$ (Equation 3).

### 5.2 Quality of service

We define another objective function, i.e. QoS provided by the network using metrics like packet delivery ratio (PDR) and end-to-end delay (E2ED). Thus, QoS is formulated as a linear sum of PDR and E2ED as follows:

$$QoS = v_1 \cdot PDR - v_2 \cdot E2ED.$$  \hspace{1cm} (5)
The weight of $PDR$, i.e. $r_1$ is kept positive as we have to improve $PDR$ in order to improve $QoS$ while, the weight of $E2ED$, i.e. $r_2$ is kept negative as we have to reduce $E2ED$ to order to improve $QoS$. Setting $r_1 = r_2 = 0.5$, we get,

$$QoS = 0.5PDR - 0.5E2ED. \quad (6)$$

### 5.3 Energy consumption ($E$)

The objective function, i.e. energy consumption by the network is evaluated using both energy consumed during UAVs’ flight as well as during data transmission. The energy consumed by a UAV during flight of $T$ seconds is defined as [33]:

$$E_F = (γ + ρ)d.T + P_{max}^s \left( \frac{b_v}{s} \right), \quad (7)$$

where $γ$ is minimum power required by a UAV to fly in air, $ρ$ is motor speed multiplier. $P_{max}^s$ and $T$ are the maximum power of the motor, speed and flight time of UAV, respectively. Thus, the total energy consumed by $N$ UAVs during their flight time of $T$ seconds is:

$$E1 = N.E_F. \quad (8)$$

The energy consumed during transmission is evaluated using simple model of wireless communication for 1-hop distance [12]. According to this model, the amount of energy spent by a source to send $m$ bits of data is defined as:

$$E_T (m, d) = \begin{cases} m.E_{elect} + m.E_{fr}.d^2 & \text{if } d < d_0 \\ m.E_{elect} + m.E_{amp}.d^4 & \text{else} \end{cases} \quad (9)$$

where $E_{elect}$ is defined as electronics energy and $E_{fr}.d^2$ or $E_{amp}.d^4$ are defined as amplifier energies which depend on the distance ($d$) between source and destination. The threshold distance, i.e. $d_0$ is calculated as:

$$d_0 = \sqrt{\frac{E_{fr}}{E_{amp}}}. \quad (10)$$

The amount of energy dissipated by a destination node to receive $m$ bits of data is formulated as:

$$E_R (m) = m.E_{elect}. \quad (11)$$

On the basis of this wireless communication model, the total energy consumed by a network while transmitting data for a period of $T$ seconds is calculated as $E2$. Now, the total energy consumed by an AMN for a period of $T$ seconds is defined as:

$$E = E1 + E2 \quad (12)$$

### 5.4 Equal load distribution

In order to achieve novel network objective, i.e. equal load distribution, we try to assure that each homogenous UAV should cover equal number of targets. The equal distribution maintains a balance of resource utilization between UAVs’ so that each and every UAV gets equally utilized. To achieve equal load distribution of the network, we define load deviation of the network (LDN) as the average of the square difference of the number of targets covered by each UAV $u \in U$ from the average number of targets that should be assigned to each UAV. It is formulated as:

$$LDN = \frac{\sum_{u=1}^{N} (CO_u - ANT)^2}{N},$$

where

$$ANT = \frac{M}{N}. \quad (13)$$

where $CO_u$ is the number of targets covered by a UAV $u \in U$, $M$ is the total number of targets and $N$ is the total number of UAVs and $ANT$ is the average number of targets that should be assigned to each UAV of an AMN for equal load distribution. The minimization of $LDN$ ensures more equal load distribution over all UAVs.

### 5.5 Network fault tolerance (NFT)

The above-defined objectives primarily deal with the connection between UAVs and targets. However, there is also a need to focus on the connection between UAVs and the connection between UAVs and the control station to improve the network performance. Reina et al. [10] focused on the connection between UAVs to improve the fault tolerance of the network. They focused on maximizing the size of minimum vertex cuts in the network to keep the UAVs connected for longer time. The reason to consider minimum vertex cut is that it is the smallest vertex set (or set of UAV nodes) whose removal can disconnect the whole network.

We further contributed to this direction by minimizing the number of minimum vertex cuts in the network. This in turn will reduce the number of such vertex sets in the network whose removal can disconnect the whole network, and thus, will keep the network connected for longer time. Moreover, we focused on increasing the number of UAVs connected to the control station so that the control station remains connected with the network of UAVs for longer time. Now, we combined all these factors into the objective function, i.e. NFT, which considers:

(i) Maximizing the size of minimum vertex cuts in the network, denoted as $m$.

(ii) Minimizing the number of minimum vertex cuts in the network, denoted as $k$. 

$$NFT = \max\{m, k\}$$
(iii) Maximizing the number of UAVs connected with the control station, denoted as \( dd \).

Thus, the NFT improves the connection between UAVs as well as the connection between UAVs and the control station to keep the UAVs and the control station connected with each other for longer time. We defined it as a linear sum of the size of minimum vertex cuts \( (m) \), number of minimum vertex cuts \( (k) \) and the number of UAVs connected with the control station \( (dd) \) as follows:

\[
NFT = u1 \times m - u2 \times k + u3 \times dd. \tag{14}
\]

The \( u1, u2 \) and \( u3 \) are the weights of \( m, k \) and \( dd \), respectively, which represents their respective contribution to the NFT. The sign with \( m \) and \( dd \) is positive as we have to increase \( m \) and \( dd \) to improve \( NFT \), while the sign with \( k \) is negative as we have to decrease \( k \) to improve \( NFT \).

Here, we assume that increasing size of minimum vertex cuts is more important than minimizing number of minimum vertex cuts and maximizing number of UAVs connected to the control station. This is because the minimum vertex cut can break the whole network, therefore, its size need to be increased to maintain the network connectivity. Therefore, we kept \( u1 = 0.5 \) while \( u2 = u3 = 0.25 \), such that \( u1 + u2 + u3 = 1 \). Thus, Equation \( (14) \) becomes:

\[
NFT = 0.5 \times m - 0.25 \times k + 0.25 \times dd. \tag{15}
\]

### 5.6 Fitness function

Now, we converted the above five network objectives, i.e. \( TC, QoS, E, LDN \) and \( NFT \) into a single objective as the optimization algorithms can optimize only a single objective. Therefore, a single objective is created by the linear combination of \( TC, QoS, E, LDN \) and \( NFT \) as follows:

\[
Utility = w1 \times TC + w2 \times QoS - w3 \times E - w4 \times LDN + w5 \times NFT. \tag{16}
\]

The significance of \( Utility \) is to measure the quality of UAVs’ deployment while forming an AMN. Here, we have to maximize the fitness function \( Utility \), therefore, the sign of \( TC, QoS \) and \( NFT \) is positive while the sign of \( E \) and \( LDN \) is negative. Setting \( w1 = 0.4 \) (as target coverage is comparatively more significant than other objective) and \( w2 = w3 = w4 = w5 = 0.15 \), Equation \( (16) \) becomes:

\[
Utility = 0.4 \times TC + 0.15 \times QoS - 0.15 \times E - 0.15 \times LDN + 0.15 \times NFT. \tag{17}
\]

Based on Equation \( (17) \), the network’s \( Utility \) value is maximized by an optimization algorithm to find the optimal UAVs’ positions. However, to make all objectives to contribute equally, we normalized each objective function within the range of \([0, 1]\) using the following formula \([34]\):

\[
f_i^N(x) = \frac{f_i(x) - \min \{f_i\}}{\max \{f_i\} - \min \{f_i\}}.
\]

### 6 EXPERIMENTAL ANALYSIS

In this section, the rigorous experiments are performed by the optimization algorithms and their results are analysed to find the best one each scenario.

#### 6.1 Experimental requirements

The system requirements to run the experiments are described as follows: Ubuntu 16.04 OS, 4 GB memory, 500 GB hard disk and Intel Core i5 processor. The software requirements are as follows: ns-2.35 network simulator, for simulating the network environment; MATLAB, for executing the optimization algorithms; Minitab (https://www.minitab.com), for performing different analysis over the collected results. The network simulator generates trace file while simulating the network. The results of the trace file are used to obtain PDR, E2ED and energy consumption of the network.

#### 6.2 Test scenarios

In the test scenario, we assumed a disaster struck area \( A \) of size \( 1000 \times 1000 \text{ m}^2 \) where \( N = 30 \) UAVs have been deployed to facilitate communication to \( M \) number of disconnected targets. To simulate this scenario, we used ns-2 simulator where we have set the values of other simulation parameters according to Table 1. The wireless signals between UAVs and targets suffer from fading effect of the objects that lie between them, therefore, shadowing propagation model is used in this scenario. Furthermore, to allow data transmission, physical layer uses IEEE 802.11s between UAVs in order to support mesh topology between them and IEEE 802.11 g between UAVs and targets in order to support Wi-Fi communication between them. The routing layer uses Geographic Greedy Forwarding (GGF) and transport layer uses User Datagram Protocol (UDP). The flow of data generated by targets and control station is assumed to be constant with packet size of 1500 bytes and CBR rate of 10 Mbps for each of the 10 sessions. This whole scenario is simulated for a period of 60 seconds.

In addition to this, the values of different parameters of UAVs are set as follows: transmission range is set to 250 m, visibility angle is set to 60° and initial energy of each UAV is set to 100 kJ. The parameters that effect energy consumed during UAVs flight, i.e. \( \rho, \gamma, P_{\text{max}} \) and \( s \) are set to 1, 30, 85 Watt and 2 m/s, respectively. Furthermore, the parameters that effect energy consumed during data transmission, i.e. \( E_{\text{elec}}, E_{\text{fr}} \) and
### Table 1: Simulation parameters

| Parameter             | Value                      |
|-----------------------|----------------------------|
| Target area           | 1000 × 1000 m²             |
| Number of UAVs        | 30                         |
| Propagation model     | Shadowing                  |
| PHY/MAC layer         | IEEE 802.11s/g             |
| Routing layer         | GGF routing protocol       |
| Transport layer       | UDP                        |
| CBR packet size       | 1500 bytes                 |
| CBR data flow         | 10 sessions                |
| CBR rate              | 10 Mbps                    |
| Transmission range    | 250 m                      |
| Visibility angle      | 60°                        |
| UAV’s initial energy  | 100 kJ                     |
| $P_1$                 | $30$                       |
| $\Gamma$              | $85$ Watt                  |
| $S$                   | 2 m/s                      |
| $E_{\text{elec}}$     | 50 nJ/bit                  |
| $E_{\phi}$            | 10 pfJ/bit/m²              |
| $E_{\text{amp}}$      | 0.0013 pfJ/bit/m⁴         |
| Simulation time       | 60 s                       |

$E_{\text{amp}}$ are set to 50 nJ/bit, 10 pfJ/bit/m² and 0.0013 pfJ/bit/m⁴, respectively.

Now, we create three different scenarios based on number of targets/end users in a given area. Here, if number of targets is between 10 and 40, then it is called as small-scale scenario. If number of targets is between 50 and 80, then it is called as a medium scale and if number of targets is between 90 and 120, then it is called as a large-scale scenario. These different scenarios are presented in Table 2.

### Table 2: Different test scenarios

| Scenarios | Problems | Problem size (Number of targets) |
|-----------|----------|----------------------------------|
| Small     | TP-1     | 10                               |
|           | TP-2     | 20                               |
|           | TP-3     | 30                               |
|           | TP-4     | 40                               |
| Medium    | TP-5     | 50                               |
|           | TP-6     | 60                               |
|           | TP-7     | 70                               |
|           | TP-8     | 80                               |
| Large     | TP-9     | 90                               |
|           | TP-10    | 100                              |
|           | TP-11    | 110                              |
|           | TP-12    | 120                              |

represents an orthogonal array suggested by Taguchi’s DoE. For GA, HS, SA, WWO and HGA-SA, $L_{25}$ orthogonal array is suggested; therefore, 25 experiments are performed. For HWWO-HSA, $L_{27}$ orthogonal array is suggested; therefore, 27 experiments are performed to obtain the best level for each parameter.

Using different parameter combinations of an orthogonal array, the fitness function, i.e. $Utility$ (Equation 17) is obtained for each optimization algorithm which is further used to calculate its corresponding relative percentage deviation ($RPD$) using the formula:

$$ RPD = \frac{|Utility_{\text{algo}} - Utility_{\text{best}}|}{Utility_{\text{best}}} \times 100. $$

The RPD represents the percentage deviation of a solution from the best one. According to Taguchi’s DoE, the RPD values are used to calculate signal-to-noise ratio (SNR) for each parameter using formula for maximization problem as:

$$ SNR = -10\log\left(\frac{1}{n} \sum_{i=1}^{n} \frac{1}{y_i^2}\right), $$

where $n$ is the number of observations and $y_i$ is the $i$th observation value. The higher value of SNR infers higher amount of information and lesser noise, and thus, its higher value is preferred. Therefore, a parameter value having higher SNR will be more informative than the one having smaller SNR. This will ensure that a metaheuristic-based optimization algorithm is well adopted according to the given problem and thus has been tuned quite well. The SNR for different parameters of each algorithm is presented in Figure 4. On the basis of the SNR value for different parameter levels, the best level which has highest SNR value is obtained in column 8 of Table A1 (see Appendix).
6.4 Comparison of optimization algorithms

The tuned optimization algorithms run each test problem for 20 times. These 20 values of Utility generated during each run of a test problem by an optimization algorithm are averaged and presented in Table A2 (see Appendix). On the basis of these Utility values, the corresponding RPD values are calculated using Equation (19). This is so because RPD provide more significance to the results while comparison by scaling the Utility values in the range of 0–100. Furthermore, it directly shows the percentage deviation (either increase or decrease) of an optimization algorithm in comparison to the best one. Figure 5 presents RPD values by the different optimization algorithms for each test problem. These RPD values suggest that hybrid algorithms are performing quite better than originals, i.e. GA, HS, SA and WWO. Furthermore, the RPD values of HGA-SA are lower as compared to that of others in small-scale test problems, suggesting that HGA-SA is performing better than others.
The RPD values of HWWO-HSA are lower as compared to that of other in medium and large-scale test problems, suggesting that HWWO-HSA performs better than others in medium and large-scale scenarios. In order to validate the conclusions drawn from RPD values, we have performed validation test in Section 6.5.

The CT is defined as the time required for executing an algorithm for maximum number of iterations. The CT of the different optimization algorithms for each test problem is presented in Figure 6. The average CT for each algorithm is presented in Figure 7, which shows that being hybrid, the CT of hybrids is comparatively higher than that of the originals. The WWO has lowest CT while HWWO-HSA has highest CT. Furthermore, GA has second lowest CT, then HS, then SA and then HGA-SA. The CT of HWWO-HSA is 18.4% more than that of HGA-SA.

Figure 8 shows the iteration at which the optimization algorithms converge to an optimal solution. The iteration at which we decide that the convergence is achieved is the one after which there is either no change or negligible change in the optimal solution. In this figure, we can observe that the proposed HWWO-HSA converges in comparatively lower number of iterations than that of WWO. This is due to Metropolis Monte Carlo selection criteria of SA which discard random unfit solutions at later execution phase of the hybrid algorithm.

6.5 Validation test

In order to validate the conclusions drawn from Figure 5 (RPD plot), we performed a validation test. Before applying validation test, we performed normality and equal variance test. The Shapiro–Wilk’s normality test, as shown in Figure 9(a), shows that all the residuals are not aligned to the straight diagonal line. This suggests that RPD values are not normally distributed. Furthermore, Levene’s equal variance test, as shown in Figure 9(b), shows that confidence interval of standard deviation of all the algorithms overlap with each other. This suggests that RPD values generated by each optimization algorithm have equal variance.

In such a situation where data set is not normally distributed (i.e. non-parametric data) but have equal variance, KW test can perform well. The KW test compares the median values of different samples. If the $p$-value of KW test is less than the significance level, $\alpha = 0.05$, then with 95% confidence it suggests that there is a significant difference in median values between at least two algorithms. In all scenarios, it is observed that the $p$-value < 0.05, as shown in column 6 of Table A3 (see Appendix), which suggests that the results are significant. Furthermore, the confidence interval plots for RPD having median RPD value (blue circle in Figure 10) for each optimization algorithm in different scenarios are presented in Figure 10.
basis of the confidence interval plots, it is observed that the proposed HGA-SA has the lowest median RPD value than others in small-scale scenarios while in medium and large-scale scenarios, the proposed HWWO-SA has the lowest median RPD value than others. The median RPD values of each algorithm in different test problems are presented in column 3 of Table A3 (see Appendix).

On the basis of KW test, it has been suggested that the results are significant and at least two algorithms have different median RPDs in all the test scenarios. Now, to find which pairs of algorithms are different we have performed Dunn’s post-hoc test. This post-hoc test finds out the significance difference in median RPD between each pair of algorithms as shown in column 5 of Table A3 (see Appendix). On the basis of this, we can say that median RPD of HGA-SA has significant difference with that of GA and HS in small-scale scenarios. This suggests that in small scale, the average utility of HGA-SA is 11.01% and 12% more than that of GA and HS, respectively. Further in medium scale, HWWO-HSA has significant difference with GA, HS, SA and WWO, while HGA-SA has significant difference with WWO. This infers that in medium scale, the average utility of HWWO-HSA is 12.7%, 6.5%, 15.3% and 23.2% more than that of GA, HS, SA and WWO, respectively, while HGA-SA is 17.2% more than that of WWO. In large scale, HWWO-HSA has significant difference with GA, HS and SA. This suggests that in large scale, the average utility of HWWO-HSA is 7.08%, 11.18% and 9.86% more than that of GA, HS and SA, respectively.

7 | HARDWARE AND SOFTWARE CHALLENGES

According to Figure 1, the end devices communicate with the control station with the help of an AMN. An AMN is formed using UAVs that mount mesh routers over them to provide data transmission between end devices and the control station.
Furthermore, VoIP server is used as a framework which enables voice over IP services between communicating devices. The hardware and software requirements for our system model are presented below:

7.1 | End devices

- The end devices can be Android Phone, iPhone, wired or wireless phones, PC (Windows, Linux or Mac)
- The software installed over end device is Zoiper (https://www.zoiper.com/) which provides session initiated protocol (SIP)-based calling between the control station and end devices. The Zoiper associates an extension number to each end device present in the asterisk network handled through VoIP server.

7.2 | VoIP server

The VoIP server facilitates communication between end devices and control station by providing a communication framework between them. The server enables voice communication through packet transmission within the network framework.

- The hardware for VoIP server can be Linux-based system (PC or Arduino or Raspberry Pi) with wireless connectivity (IEEE 802.11)
- The software for VoIP server can be an asterisk open source communication software (https://www.asterisk.org). Furthermore, FreePBX open source (https://www.freepbx.org) which is a web-based GUI is used to manage the asterisk framework.

7.3 | Mesh routers

- The hardware for mesh routers can be either light weight router with wireless connectivity or Arduino/Raspberry Pi with wireless connectivity module (running over Linux OS). These routers are mounted over UAV.
- The wireless routers can get configured using their own manufactures web-based configuration software. However, for Arduino or Raspberry Pi with Linux OS, we need to install OpenWRT (https://wiki.openwrt.org) firmware which makes them behave like a wireless access point, i.e. wireless router through which the end devices get connected to the AMN.

7.4 | Control station

The control station should also be configured with the VoIP server so that it can connect to the AMN and able to communicate to the end devices.

7.5 | Unmanned aerial vehicles

The major concern while selecting UAVs according to the application requirements are as follows:

- Cost of UAVs
- Remote control range of UAVs
- The range of the wireless router mounted over UAVs
- Battery support for UAVs
- Take-off weight capacity for UAVs

8 | CONCLUSION AND FUTURE SCOPE

This paper deals with the optimal placement of the available number of UAVs to provide temporary communication in a post-disaster scenario. To optimize placement of UAVs, the different objectives like target coverage, QoS, equal load distribution, fault tolerance and energy consumption along with maintaining connectivity within the network have been considered. Metaheuristic-based optimization algorithms provide sub-optimal solutions for such complex defined problems. The optimization algorithms used here are GA, HS, SA and WWO and their proposed hybrids, i.e. HGA-SA and HWWO-HSA. Furthermore, the performance of these hybrid algorithms is improved by tuning their parameters using Taguchi's DoE. The tuned hybrid algorithms are executed in different test scenarios to generate the results. The test scenarios are categorized as small scale, medium scale and large scale based on the number of targets in a given region. The RPD values are used to compare the outputs generated by hybrid algorithms with the original ones. Furthermore, to validate the results drawn from the RPD comparison, we used a non-parametric test, i.e. KW test.

On the basis of the KW test, it is observed that the proposed hybrids are performing better than the original ones. Moreover, HGA-SA has the lowest median RPD value as compared to that of others in small-scale scenarios while HWWO-HSA has the lowest median RPD value as compared to that of others in both medium and large-scale scenarios. Also, the p-values of the KW test are less than the significance level in each of the test scenarios, suggesting that the results are significant. Furthermore, Dunn's post hoc test is used to find which pairs of optimization algorithms differ significantly. The test results suggest that in small scale the proposed hybrid HGA-SA significantly performs 11.01% and 12% better than GA and HS, respectively. In medium scale, the proposed hybrid HWWO-HSA significantly performs 12.7%, 6.5%, 15.3% and 23.2% better than GA, HS, SA and WWO, respectively, while hybrid HGA-SA performs 17.2% better than WWO. Moreover, in large scale, HWWO-HSA significantly performs 7.08%, 11.18% and 9.86% better than GA, HS and SA, respectively. However, the CT of hybrids is comparatively higher than that of originals. The CT of HWWO-HSA is 18.4% more than that of HGA-SA.

In future, we may consider implementing this model in a disaster-struck region having dynamic targets. Moreover, other objectives like interference, replacement of dead UAVs, battery
charging, etc. can be considered for improving the performance of AMNs. Also, we can apply tuned metaheuristic-based multi-objective optimization algorithms in our designed problem to analyse their results.

REFERENCES

1. Hayat, S., Yarmaz, E., Muzaffar, R.: Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. IEEE Commun. Surv. Tutorials 18(4), 2624–2661 (2016)
2. Ilouzid, C., Clemente, C., Soraghan, J: GNSS-based passive UAV monitoring: A feasibility study. IET Radar, Sonar Navig. 14(4), 516–524 (2019)
3. Chen, S., et al: Accurate landslide detection leveraging UAV-based aerial remote sensing, IET Commun. 14(15), 2434–2441 (2020)
4. Di Felice, M., et al: Self-organizing aerial mesh networks for emergency communication. In: IEEE 25th Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC), pp. 1631–1636. IEEE (2014)
5. Fan, B., et al: Review on the technological development and application of UAV systems. Chin. J. Electron. 29(2), 199–207 (2020)
6. Mazzarella, J.R., Wengrovitz, M.S., Mutualink Inc.: System and method for dynamic wireless aerial mesh network. US Patent 9,654,200, 2017
7. Malandrino, F., et al: Planning UAV activities for efficient user coverage in disaster areas. Ad Hoc Networks 89, 177–185 (2019)
8. Sabino, S., Horta, N., Grilo, A.: Centralized unmanned aerial vehicle mesh network placement scheme: A multi-objective evolutionary algorithm approach. Sensors 18(12), 4387 (2018)
9. Callouet, C., Razafindralambo, T: Efficient deployment of connected unmanned aerial vehicles for optimal target coverage. In: Global Information Infrastructure and Networking Symposium (GIIS), pp. 1–8. IEEE (2017)
10. Reina, D.G., Tawfik, H., Toral, S.L.: Multi-subpopulation evolutionary algorithms for coverage deployment of UAV-networks. Ad Hoc Networks 68, 16–32 (2018)
11. Younis, M., Akkaya, K.: Strategies and techniques for node placement in wireless sensor networks. A survey. Ad Hoc Networks 6(4), 621–655 (2008)
12. Gupta, M., Varma, S.: Optimal placement of UAVs of an aerial mesh network in an emergency situation. J. Ambient Intell. Hum. Comput. 1–16 (2020)
13. Taguchi, G.: Introduction to quality engineering: designing quality into products and processes (No. 658.562 T3), (1986)
14. Mazzarella, J.R., Wengrovitz, M.S., Mutualink Inc.: System and method for dynamic wireless aerial mesh network. US Patent 9,654,200, 2017
15. Žulj, I., Kramer, S., Schneider, M.: A hybrid of adaptive large neighborhood search and tabu search for the order-batching problem. Eur. J. Oper. Res. 264(2), 653–664 (2018)
16. Senel, E.A., et al: A novel hybrid PSO–GWO algorithm for optimization problems. Eng. Comput. 35(4), 1359–1373 (2019)
17. Murthy, T.S., Roy, M.S., Varma, M.K.: Improving the performance of association rules hiding using hybrid optimization algorithm. J. Appl. Secur. Res. 15, 1–15 (2020)
18. Mamaghani, E.J., Prins, C., Chen, H.: A hybrid algorithm for collaborative transportation planning among carriers. Int. J. Econ. Manage. Eng. 12(3), 383–391 (2018)
19. Huang, W., Lin, W., Xu, S.: Application of graph theory and hybrid GA-SA for operation sequencing in a dynamic workshop environment. Comput. Aided Des. Appl. 14(2), 148–159 (2017)
20. Anto, S., Chandramathi, S., Femina, B: An expert system based on SVM and hybrid GA-SA optimization for disease diagnosis. Int. J. Biol. Biomed. Eng. 2, 86–97 (2017)
21. Thai, M.T.et al: Coverage problems in wireless sensor networks: Designs and analysis. Int. J. Sem. Netw. 3(3), 191–200 (2008)
22. Sabino, S., Grilo, A.: Topology control of unmanned aerial vehicle (UAV) mesh networks: A multi-objective evolutionary algorithm approach. In Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, pp. 45–50 (2018)
23. Chen, Y., Zhang, H., Xu, M.: The coverage problem in UAV network: A survey. In Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1–5. IEEE (2014)
24. Hu, L., et al: Placement optimisation method for multi-UAV relay communication. IET Commun. 14(6), 1005–1015 (2020)
25. Singh, G., et al: Hybridization of wave water optimization and sequential quadratic programming for cognitive radio system. Soft Comput. 23(17), 7991–8011 (2019)
26. Sahebjamnia, N., Fathollahi-Fard, A.M., Hajighaie-Keshmeti, M: Sustain- able tire closed-loop supply chain network design: Hybrid metaheuristic algorithms for large-scale networks. J. Cleaner Prod. 196, 273–296 (2018)
27. Rong, Z.Y., et al: A hybrid evolutionary algorithm for combined road-rail emergency transportation planning. In International Conference on Swarm Intelligence, pp. 465–476. Springer, Cham (2018)
28. Chouhan, V.K., et al: Multi-facility-based improved closed-loop supply chain network for handling uncertain demands. Soft Comput. 24, 7125–7147 (2020)
29. Holland, J.H.: Genetic algorithms. Sci. Am. 267(1), 66–73 (1992)
30. Greem, Z.W., Kim, J.H., Loganathan, G.V: A new heuristic optimization algorithm: Harmony search. Simulation 76(2), 60–68 (2001)
31. Aarts, E., Korst, J.: 1988. Simulated Annealing and Boltzmann Machines, (1989), John Wiley & Sons, pp. 3–75
32. Zheng, Y.J.: Water wave optimization: A new nature-inspired metaheuristic. Comput. Oper. Res. 55, 1–11 (2015)
33. Pugliese, L.D.P ., et al: Modelling the mobile target covering problem using flying drones. Optim. Lett. 10(5), 1021–1052 (2016)
34. Coello, C.A.C., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, vol. 5. Springer, New York. pp. 79–104 (2007)

How to cite this article: Gupta M, Varma S. Metaheuristic-based optimal 3D positioning of UAVs forming aerial mesh network to provide emergency communication services. IET Commun. 2021;15: 1297–1314. https://doi.org/10.1049/cm.212112
## APPENDIX

### TABLE A1  The three-level assignment for different parameters with their best level

| Algorithms | Factors | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Best level |
|------------|---------|---------|---------|---------|---------|---------|------------|
| GA         | $P_c$   | 0.6     | 0.65    | 0.7     | 0.75    | 0.8     | 0.65       |
|            | $P_m$   | 0.1     | 0.125   | 0.15    | 0.175   | 0.2     | 0.175      |
|            | $P$     | 50      | 75      | 100     | 125     | 150     | 150        |
|            | $MaxIt$ | 200     | 250     | 300     | 350     | 400     | 200        |
| HS         | $HCIR$  | 0.8     | 0.825   | 0.85    | 0.875   | 0.9     | 0.875      |
|            | $PAR$   | 0.1     | 0.125   | 0.15    | 0.175   | 0.2     | 0.15       |
|            | $P$     | 50      | 75      | 100     | 125     | 150     | 125        |
|            | $MaxIt$ | 200     | 250     | 300     | 350     | 400     | 300        |
| SA         | $alpha$ | 0.85    | 0.875   | 0.9     | 0.925   | 0.95    | 0.95       |
|            | $T$     | 700     | 800     | 900     | 1000    | 1100    | 700        |
|            | $SubIt$ | 20      | 25      | 30      | 35      | 40      | 20         |
|            | $MaxIt$ | 1000    | 1100    | 1200    | 1300    | 1400    | 1100       |
| WWO        | $h_{max}$ | 5     | 5.25    | 5.5     | 5.75    | 6      | 5          |
|            | $k_{max}$ | 10   | 11      | 12      | 13      | 14     | 10         |
|            | $B$     | 0.001   | 0.0025  | 0.005   | 0.0075  | 0.01   | 0.01       |
|            | $M$     | 1.001   | 1.0025  | 1.005   | 1.0075  | 1.01   | 1.0025     |
|            | $P$     | 30      | 40      | 50      | 60      | 70     | 60         |
|            | $MaxIt$ | 200     | 250     | 300     | 350     | 400    | 300        |
| HGA-SA     | $P_c$   | 0.6     | 0.65    | 0.7     | 0.75    | 0.8    | 0.6        |
|            | $P_m$   | 0.1     | 0.125   | 0.15    | 0.175   | 0.2    | 0.175      |
|            | $T$     | 700     | 800     | 900     | 1000    | 1100   | 800        |
|            | $alpha$ | 0.85    | 0.875   | 0.9     | 0.925   | 0.95   | 0.9        |
|            | $P$     | 50      | 75      | 100     | 125     | 150    | 100        |
|            | $MaxIt$ | 200     | 250     | 300     | 350     | 400    | 300        |
| HWWO-HSA   | $HCIR$  | 0.8     | 0.85    | 0.9     | –       | –      | 0.85       |
|            | $PAR$   | 0.1     | 0.15    | 0.2     | –       | –      | 0.2        |
|            | $h_{max}$ | 5     | 5.5     | 6      | –       | –      | 6          |
|            | $k_{max}$ | 10   | 12      | 14     | –       | –      | 10         |
|            | $B$     | 0.001   | 0.005   | 0.01   | –       | –      | 0.001      |
|            | $M$     | 1.001   | 1.005   | 1.01   | –       | –      | 1.01       |
|            | $alpha$ | 0.85    | 0.9     | 0.95   | –       | –      | 0.95       |
|            | $T$     | 800     | 900     | 1000   | –       | –      | 900        |
|            | $P$     | 30      | 50      | 70     | –       | –      | 50         |
|            | $MaxIt$ | 200     | 300     | 400    | –       | –      | 300        |
### TABLE A2
The utility and RPD values of the optimization algorithms in different test problems

| Algorithms | Test problems | Utility | RPD | Utility | RPD | Utility | RPD | Utility | RPD | Utility | RPD | Utility | RPD |
|------------|---------------|---------|-----|---------|-----|---------|-----|---------|-----|---------|-----|---------|-----|
| GA         | TP-1          | 0.2884  | 7.593 | 0.2562  | 17.91 | 0.2869  | 8.074 | 0.3121  | 3.684 | 0.3006  | 11.674 | 0.3006  | 11.674 |
|            | TP-2          | 0.2713  | 8.127 | 0.2771  | 6.163 | 0.2953  | 0     | 0.2827  | 4.267 | 0.2891  | 2.099 | 0.2921  | 1.083 |
|            | TP-3          | 0.2807  | 10.462 | 0.2799 | 10.717 | 0.2871 | 8.421 | 0.3018 | 3.732 | 0.3135 | 0 | 0.2769 | 11.674 |
|            | TP-4          | 0.2622  | 15.31 | 0.2801  | 9.528 | 0.284  | 8.268 | 0.3075  | 0 | 0.2906  | 0 | 0.2793 | 9.786 |
|            | TP-5          | 0.2719 | 13.49 | 0.2887  | 8.145 | 0.2757  | 12.281 | 0.2562 | 18.485 | 0.3143 | 0 | 0.3004 | 4.422 |
|            | TP-6          | 0.2827  | 8.303 | 0.2915 | 5.449 | 0.2776 | 9.957 | 0.2507 | 18.683 | 0.2842 | 7.817 | 0.3083 | 0 |
|            | TP-7          | 0.2612  | 16.496 | 0.2879 | 7.96 | 0.2751 | 12.052 | 0.2253 | 27.973 | 0.2796 | 10.613 | 0.3128 | 0 |
|            | TP-8          | 0.2768  | 10.738 | 0.2916 | 5.965 | 0.2183 | 29.603 | 0.2676 | 13.705 | 0.2938 | 5.256 | 0.3101 | 0 |
|            | TP-9          | 0.2897  | 5.109 | 0.2648 | 13.265 | 0.2844 | 6.845 | 0.2889 | 5.371 | 0.2852 | 6.383 | 0.3053 | 0 |
|            | TP-10         | 0.2736  | 9.94 | 0.2776 | 8.624 | 0.2663 | 12.343 | 0.3038 | 0 | 0.293 | 3.554 | 0.2931 | 3.522 |
|            | TP-11         | 0.2774  | 7.409 | 0.2572 | 14.152 | 0.2781 | 7.176 | 0.2686 | 10.347 | 0.2922 | 2.469 | 0.2996 | 0 |
|            | TP-12         | 0.2884  | 7.237 | 0.2877 | 7.462 | 0.2715 | 12.673 | 0.2719 | 12.544 | 0.2939 | 5.467 | 0.3109 | 0 |

### TABLE A3
Each optimization algorithm’s median RPD, mean RPD with std. deviation, significant difference with other algorithms and p-value of KW test in different test scenarios

| Scenario | Algorithm | Median RPD | Mean RPD ± std | Significance with | p-Value |
|----------|-----------|------------|----------------|--------------------|---------|
| Small    | GA        | 9.294      | 10.373 ± 3.519 | HGA-SA             | .029    |
|          | HS        | 10.123     | 11.080 ± 4.946 | HGA-SA             |         |
|          | SA        | 8.172      | 6.191 ± 4.130  |                    |         |
|          | WWO       | 3.899      | 4.012 ± 2.740  |                    |         |
|          | HGA-SA    | 0.000      | 0.525 ± 1.050  | GA, HS             |         |
|          | HWWO-HSA  | 6.735      | 6.557 ± 4.994  |                    |         |
| Medium   | GA        | 12.114     | 12.257 ± 3.532 | HWWO-HSA           | .002    |
|          | HS        | 6.963      | 6.880 ± 1.372  | HWWO-HSA           |         |
|          | SA        | 12.166     | 15.974 ± 9.146 | HWWO-HSA           |         |
|          | WWO       | 18.587     | 19.712 ± 5.969 | HGA-SA, HWWO-HSA   |         |
|          | HGA-SA    | 6.536      | 5.922 ± 4.514  | WWO                |         |
|          | HWWO-HSA  | 0.000      | 1.106 ± 2.211  | GA, HS, SA, WWO    |         |
| Large    | GA        | 7.323      | 7.424 ± 1.977  | HWWO-HSA           | .014    |
|          | HS        | 10.944     | 10.876 ± 3.325 | HWWO-HSA           |         |
|          | SA        | 9.76       | 9.750 ± 3.180  | HWWO-HSA           |         |
|          | WWO       | 7.859      | 7.066 ± 5.585  |                    |         |
|          | HGA-SA    | 4.511      | 4.519 ± 1.852  |                    |         |
|          | HWWO-HSA  | 0.000      | 0.881 ± 1.761  | GA, HS, SA         |         |

Bold values represent the lower RPD values which are preferred.