ECoR: Energy-Aware Collaborative Routing for Task Offload in Sustainable UAV Swarms

Anandarup Mukherjee, *Student Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Vadde Santosha Pradeep Chandra, and Narendra Singh Raghuwanshi

Abstract—In this work, we propose an Energy-aware Collaborative Routing (ECoR) scheme for optimally handling task offloading between source and destination UAVs in a grid-locked UAV swarm. We divide the proposed scheme into two parts – routing path discovery and routing path selection. The scheme selects the most optimal path between a source and destination from a massive set of all possible paths, based on the maximization of residual energy of UAVs along a selected path. This routing path selection ensures balanced energy utilization between members of the UAV swarm and enhances the overall path lifetime without incurring additional delays in doing so. Actual readings from our small-scale UAV swarm testbed are utilized to emulate a large-scale scenario and analyze the behavior of our proposed scheme. Upon comparison of the ECoR scheme with broadcast-based routing and the shortest path based routing, we observe better sustainability regarding the longevity of the UAV lifetimes in the swarm, optimized individual UAV, as well as reduced collective path-based energy consumption, all the while having comparable transmission delays to the shortest path based scheme.

Index Terms—UAV swarm, aerial mesh network, Internet of Things, routing, shortest path, broadcast, path selection, collaborative processing.

1 Introduction

A swarm of Unmanned Aerial Vehicles (UAVs) is a collection of an intercommunicating group of UAVs, which are capable of performing assigned tasks collaboratively. However, despite developments in low-power processing and sensing solutions, UAVs, especially multi-rotor types, rapidly deplete their energy sources to maintain this platform airborne. It is mainly due to the energy constraints, which dictates the flight time of UAVs, the collaborative task completion ability of the UAV swarms proves beneficial for accomplishing assigned tasks without the need for repeated charging of the UAVs' energy sources.

This paper envisions a scenario, which makes use of a homogeneous and collaborative swarm of UAVs for task accomplishment. We consider a $2D$ aerial grid space parallel to the ground plane, where we assign each grid location to a member of the homogeneous UAV swarm. Each UAV acts as a communication gateway between a user on the ground (if any) and the airborne swarm. In the majority of the present-day UAV swarms, a central ground-server acts as a communication gateway between UAVs in the swarm and a user.

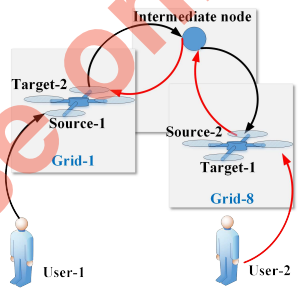


Figure 1: Schematic of the operational overview of the problem statement.

In this work, we aim to relocate and redistribute the functioning of a ground-based central server to the aerial swarm itself. Towards this, we propose the collaborative utilization of the UAVs themselves for receiving and attaching tasks in a distributed manner while they are airborne. This approach also intends to reduce the networking infrastructure required to maintain such an aerial setup, reduce deployment costs, and make the UAV swarm solution highly robust and easy to deploy. Tasks refer to any command to a UAV, which necessitates an actuation-based response from the UAV to which the task is assigned. These commands and actions are independent of the UAV flight controls, and consists of activities elicited from the UAV sensors. As such, we propose that any UAV (node) in the swarm is eligible to receive a task from a user in its communication range and forward the task details to the intended destination UAV, which is similar to a UAV-based message ferrying network [1] as shown in Fig. 1.

The primary challenge of collaborative task offload between the source (sN) and destination UAVs (dN) in a collaborative aerial UAV swarm is routing, for which we propose the **E**nergy-aware **Co**llaborative **R**outing (ECoR) scheme. ECoR consists of two stages of operation – 1) *Routing Path Discovery*, and 2) *Routing Path Selection* – before a request from the user can be forwarded to dN, and the targeted UAV sends back the data to the user. The *Routing Path Discovery* enables the sN to locate all possible paths between itself and the intended dN, whereas the *Routing*

A. Mukherjee, S. Misra, and V. S. P. Chandra are with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, India.
N. S. Raghuwanshi is with the Agriculture and Food Engineering Department, Indian Institute of Technology Kharagpur, India.

Table of notations used

Notation	Description
a	Dimension of each grid cell
$N \times N$	Number of grid cells in the deployment area
n	Number of UAVs in a selected path
t	Observation time
k	Path updation time
T	Task completion time
sN, dN	Source UAV, Destination UAV
G	Unidirectional graph formed by UAVs as nodes of G
V	Set of nodes of G, which have an adjacency list defining the edges
pL	Set of all possible paths between sN and dN
p	A selected path between sN and dN
cN	Current node/UAV
e_d	Energy depletion rate of UAV
$E_i(t)$	Energy of the i^{th} UAV at time t
$E_i(J)$	Energy required by the i^{th} UAV to complete the assigned tasks
$E(N_i)$	Energy required by the i^{th} UAV to keep it airborne
$U_p(t)$	Energy-based path score function
$R_{Ei}(t)$	Residual energy of i^{th} UAV
$TE(t)$	Cumulative residual energy of a selected routing path ($= \sum_{i=1}^{n} R_{Ei}(t)$)
$J_i(t)$	Task list assigned to the i^{th} UAV
$TJ(t)$	Cumulative task list of a selected routing path ($= \sum_{i=1}^{n} J_i(t)$)

Path Selection allows the selection of the most optimum route based on the minimization of our proposed path score function. The formulated score value for each of the available paths between sN and dN helps in selecting the routing path with the maximum collective residual energy of the constituent UAVs of the selected routing path.

Fig 1 shows user-1 located in grid-1 requesting access to a UAV in grid-8, via the UAV in its grid. The first UAV to come in contact with the user's request is designated as sN. This sN (source-1) forwards the user's request to the intended dN (target-1) via multiple hops through various UAVs, which form part of the aerial swarm. Along the same lines, when a user in grid-8 (user-2 in Fig. 1) requests access to another UAV outside its grid, the dN for user-1 now becomes the sN for user-2.

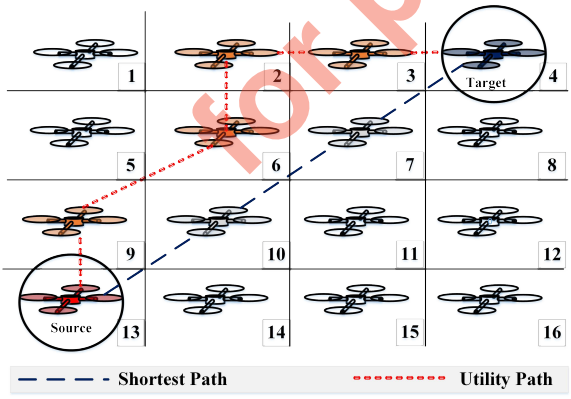


Figure 2: Layout representing the placement of UAVs in grids and operational outline of the shortest path scheme and our proposed scheme.

We consider a grid location of dimensions $a \times a$ square units such that each grid location has a UAV present in it. The maximum distance between the two extremities of the grid location is $\sqrt{2}a$. Moreover, as shown in Fig. 2 based on geometry, each UAV in a grid location can communicate directly with its 8 immediate neighbors. In continuation, the maximum distance of communication between two UAVs in separate neighboring grids is $2\sqrt{2}a$, which is also the maximum communication radius of each UAV in our scheme. As we consider a relative unit of measurement, this approach can be easily scaled-up to include larger distances depending on the communication radio in use.

A task communicated to a UAV gateway may not always be intended for it. A UAV to which the task is communicated from the ground is considered as the source UAV (sN). The sN needs to find a path to communicate with the intended UAV (target) dN, which has direct access to the task location, as shown in Figs. 1 and 2. The source and target UAVs need to maintain communication for the whole duration of the task, starting from forwarding the task details to the target UAV, the time taken to complete the task by the target UAV, and transmission of the results back to the source UAV. Under such challenging conditions, in our work, we estimate the optimal path of communication to be followed by the UAVs from sN to dN, such that every UAV node, which comprises the path does not die out due to depletion of energy at any point of time during the transmission. We formulate a collective residual energy maximization-based score function, which determines the best possible path to choose out of all the available routes. The sN retrieves these available paths, and the score function is used to evaluate them to select the most optimal path to the dN.

1.1 Contribution

The proposed work outlines a score-based optimization of task allocation in collaborative UAV swarms communicating in an ad-hoc manner. The choice of a collaborative UAV swarm enables the swarm to be robust and readily deployable without any dependency on ground-based infrastructures such a servers and computing stations. The lack of dependency also makes this swarm easily relocatable in situations requiring the immediate deployment of these solutions, such as disaster management. The following contributions have been made in this work:

- A means of routing path discovery in collaborative UAV swarms is proposed.
- A direction-based modification of the path discovery method is proposed, which is responsible for significantly reducing the processing overheads and discovery time in resource-constrained UAVs constituting the swarm.
- A residual energy maximization based score minimization approach for routing path selection between two UAVs in the swarm (ECoR) is proposed, which ensures the survival of the multi-hop path till completion of an assigned task at a destination UAV.
- Actual hardware metrics from our small-scale UAV swarm is utilized to emulate a large-scale network to analyze the performance of our approach.

2 Related Work

Works being pursued in the domain of optimal routing path selection for wired as well as wireless networks include

concepts such as energy-efficient scheduling of parallel tasks in heterogeneous computing devices [2], maximization of effective utility concerning network flow rate [3], and load balancing in wireless smart utility networks for reducing power consumption in mesh networks [4], and energy-aware task offloading in mobile cloud environments [5]. Along the same lines but in a separate domain, the work by Lawrence *et al.* addresses the problem of determining a least-cost path from a fixed start to a fixed goal position using a location-based probabilistic likelihood improvement algorithm [6]. Schemes such as these can be readily utilized for network routing based path selection too. With a focus on UAV-swarm based network routing, which has an inherent feature of high mobility and dynamic network patterns, we divide this section into two parts – 1) swarm communication architectures, and 2) routing in UAV swarms.

2.1 Communication Architectures for UAV Swarms

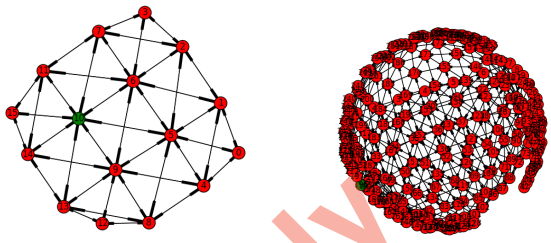
Various communication architectures exist, which deal with multi-UAV command-and-control mechanisms during flight. Mechanisms such as the leader-follower approach [7] insinuate the necessity of a single leader to communicate with the members (followers) in the UAV formation [8]. In contrast, a virtual leader topology necessitates that the whole formation is treated as a single leader, and every UAV has the same information available to them [8]. Present-day UAV swarms are either controlled by a ground station [9], [10], or undertake a self-contained command and control structure within the swarm [11]. Most of the current UAV swarm or multi-UAV network communication architectures rely on collaborative and leaderless formations [12] to protect against single-point-of-failure errors.

Despite massively reduced dependencies on ground-based infrastructure for their operation and the ability to shield against single-point errors, collaborative communication architectures require high processing power, and subsequently, spending high amounts of energy on repeated polling for updates on member status within the collaborative swarm.

2.2 Routing in UAV swarms

The highly mobile nature of UAV swarms necessitates the use of routing protocols, which are robust and dynamic enough to handle the challenges of rapidly changing network topology and the constraints of restricted energy. Bujari *et al.* comprehensively evaluate the state-of-the-art in stateless geographic routing protocols for Flying Ad-hoc NETworks (FANETs) through simulations [13]. They used the metrics of *delivery rate*, *path dilation*, and *scalability* for accessing more than twelve routing protocols for 3D networks in FANETs. Similarly, through their work Rosati *et al.* report that a Predictive Optimized Link State Routing (P-OLSR) has superior performance over OLSR concerning link performance and communication range in FANETs with dynamically changing topologies [14]. Along these lines, He *et al.* proposed an opportunistic routing protocol for FANETs [15]. Their approach jointly considered the neighbor positions and the destination direction of the UAVs or Ground Stations (GS) for implementing a course-aware routing mechanism. The various notable and standard routing approaches in FANETs include deterministic exploratory strategies [16], evolutionary approaches [17], jamming-resilient routing [18], destination-aware directional routing [19], and others.

However, most of these routing protocols seldom consider the survival of the established path between the source and the destination, all through the duration of task-driven data transmission in an energy-constrained environment.



(a) Grid size: 4×4 (b) Grid size: 16×16

Figure 3: The increase in path graph complexity for an increase in corresponding grid sizes.

3 Collaborative UAV Swarm Network

As the UAVs in the swarm form a network, which is highly ad-hoc, and the decision-making for the offload path selection is collaborative at sN, it is necessary to keep the sN updated of all the possible paths pL in the network leading up to the destination UAV.

Definition 1. ***Hop Distance:*** *It is the total number of hops required to reach a destination UAV in a selected path p from the source UAV (sN). Let a path from the i^{th} UAV to the $j+1^{th}$ UAV be $p = i, i+1, i+2, \cdots, j, j+1$, the hop distance for path p is calculated as $(j-i)$.*

In traditional networks, the path discovery is made immediately at the node at which the task/data packet resides. Additionally, the probabilistic chance of getting a forward path from an immediate node is quite high in such networks. In contrast, with limited UAVs, employing such an approach would induce additional latencies and unnecessary overheads for the energy-constrained UAVs in the network. For networked swarms, there will be prohibitive complexities involved for every coordinated task, especially if the swarm is collaborative. However, the physical upper bound on the network performance for such implementations are dictated by the number of UAVs (nodes) in the swarm and the channel bandwidth of the wireless radio being used for communication.

Definition 2. ***Task List:*** *The Task List of a i^{th} UAV is denoted as $J_i(t)$, and we define it as the total number of jobs currently assigned to UAV before time t. Ideally, the UAV should be able to complete all the tasks in its list unless an unexpected communication error occurs during its flight.*

Assumption 1. *The UAVs keep on generating data or handling data from other UAVs. This results in an increase in the number of jobs assigned to a UAV node with time.*

Algorithm 1 Path Management Score

Inputs:(G, V, sN, dN)
Output:(pL)

procedure PATHUTIL($G, sN, dN, pL, path, cN, vis$)
  Set $vis[cN]$ to $True$
  Add cN to $path$
  if $cN == dN$ **then**
    Add $path$ to pL
  else
    for `each` $node$ `in` $G[cN]$ **do**
      if $vis[node] == False$ **then**
        $pathUtil(G, sN, dN, pL, path, node, vis)$
      end if
    end for
  end if
  $path.pop()$
  Set $vis[cN]$ to $False$
end procedure

The node path update is a function of time (k) and is tunable, and can be adjusted according to the radio being used for communication between the UAV nodes. In our work, we have used Wi-Fi in our real implementation, and as per our observations, it takes a few milliseconds for all the nodes in the implemented swarm to be updated. However, with larger deployments, this period can escalate to a few seconds. Similarly, for networks with much slower data rates – Zigbee, LoRA – the update time can be significantly high (in the order of a few seconds), which necessitates a lesser frequency of network update to prevent network congestions and packet drops. However, it must be noted that the node information update frequency cannot be kept too low, as the UAV nodes are highly mobile in this ad-hoc configuration, which may result in ghost updates – by the time an update regarding a possible routing path through a set of UAVs arrive at the source node, some of the UAV might no longer be present in the network path due to depletion of their energy.

Algorithm 2 Path Discovery Algorithm

Inputs:(G, V, sN, dN)
Output:(pL)

procedure GETPATHS(G, V, sN, dN)
  Initialize vis to { }
  for `each` $node$ `in` V **do**
    `Set` $vis[node] to False$
  end for
  Initialize pL to { }
  Initialize $path$ to { }
  Initialize $currentNode$ to sN
  Call $pathUtil(G, sN, dN, pL, path, currentNode, vis)$
▷ Call Path Management Score (Algorithm 1)
  return pL
end procedure

4 Routing Path Discovery

The routing path discovery is initiated by a ground-based user under the coverage of sN (*source*) requesting task routing to dN (*destination*). Upon completion of path discovery, sN also performs the routing path selection. As our implementation is purely collaborative, there is no central controller for the UAVs in the swarm. Upon changing the sN, the path discovery and selection processes are again initiated by the new source. In our case, the physical time for exchange of path discovery information is in the order of milliseconds (ms) using a Wi-Fi-based network among the UAVs, which makes it dependent on the bandwidth of the wireless radio in use. In cases of wireless radios with low data rates such as Zigbee or LoRA, the time taken is speculated to be in the order of seconds instead of milliseconds, especially if the number of UAV nodes in the swarm is high.

We consider the formed network as an undirected graph G, where each UAV represents a node in the graph. Hence, there exists an edge between two nodes in this undirected graph if and only if the UAVs forming the nodes of G are immediate one-hop neighbors in the grid, as shown in Fig. 2. We denote the set of nodes (UAVs) as V such that each node has an adjacency list that defines its edges. The resultant set of paths between sN and dN is denoted by pL.

Algorithm 2 is modeled on a network flooding-based approach and is applied to find all the paths between sN and dN. The sN node broadcasts a packet to all its neighbors with the address of dN as the destination. As the packet moves towards dN, each UAV adds its system details to the packet and subsequently broadcasts it to all the rest of its neighbors. Once the packet reaches dN, it is sent back to sN with the path and its member node details appended to it such that $pL = \{p_1, p_2, p_3, p_4, \cdots, p_x\}$, where x is the number of possible paths between sN and dN. Unlike regular network discovery algorithms, which consider hop-based metrics, each node in our proposed approach updates its complete information – current energy level of the node $E_i(t)$, task completion time by the node T, and hop distance of the current node (cN) from the sN. This collected information of all the UAVs in each path gets updated at sN. This flooding-based approach to discover all possible paths between sN and dN, a node can be visited more than once, which results in the prohibitive increase in discovered paths, especially for higher grid numbers $N \times N$. Fig. 3 shows the increase in complexity of the connections between the neighboring nodes to a rise in swarm members. The increase in complexity directly translates to the increase in processing requirements and an increase in time for processing of information, resulting in a significant decrease in the sustainability of the UAV swarm.

4.1 Directional Routing Path Discovery

In order to induce sustainability during the routing path discovery process, we modify the algorithm to include only those nodes/UAVs/grids that are present in the direction of dN from the sN. As we restrict the UAVs within their respective grids and all UAVs in the system have information about the grid locations of all other UAVs in the swarm, the direction of dN with respect to sN is easily approximated. Unlike the naive approach (Algorithm 2), which performs

an operation analogous to flooding, and has a worst-case time complexity of $O(N^2)$ and a best-case time complexity of $\Omega(1)$, the modified version of the path discovery searches for paths that are in the approximate direction of dN as estimated from the sN resulting in an improved worst-case time complexity of $O(N)$. Denoting the grid deployment area by a matrix D, where the grid locations are the matrix cells, each with distinct line and column numbers, the swarm deployment can be represented in entirety as:

$$D = \begin{bmatrix} d_{11} & d_{12} & d^*_{13} & \dots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \dots & d_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d^*_{n2} & d_{n3} & \dots & d_{nn} \end{bmatrix}$$

Here, let us consider $sN = d^*_{n2}$ and $dN = d^*_{13}$. Using our modified path discovery approach, a mini-grid D_m is formed, which is a subset of the whole coverage grid such that it includes sN and dN at the boundary of this mini-grid as the two diagonal points of a virtual rectangle. This selection of the mini-grid can be quantified by considering $sN = d_{i,j}$ and $dN = d_{p,q}$ such that, for the indices i, j and p, q, the rows of D_m are set by $min(i, p)$ and $max(i, p)$. Similarly, the columns of D_m are set by $min(j, q)$ and $max(j, q)$. After application of the directional routing path discovery, D_m is now represented as:

$$D_m = \begin{bmatrix} d_{12} & d^*_{13} \\ d_{22} & d_{23} \\ \vdots & \vdots \\ d^*_{n2} & d_{n3} \end{bmatrix}$$

The directional paths discovered do not include UAVs that are outside this mini-grid. This measure stops the algorithm from retrieving paths that cover all the locations of the primary grid to reach dN. This directional restriction applied to the path discovery algorithm results in a significant reduction in processing and handling of the paths discovered between sN and dN. Algorithm 2 can be modified by merely incorporating the condition that $IF\ cN \in D_m,\ call\ PATH\ DISCOVERY\ ALGORITHM$. Once the paths are retrieved by sN, our proposed score function evaluates each path to find the minimum score path between sN and dN. We base this formulation on minimizing the score function of the paths $U_p(t)$ by considering the maximization of collective R_{Ei} of a path, maximizing the cumulative job list $J_i(t)$, and maximizing the number of grid cells N making up that path. The minimization of $U_p(t)$ ensures balanced use of most of the UAVs in the swarm, which would not have been possible using approaches such as shortest path-based routing.

Definition 3. ***Shortest Path:*** *We define the shortest path between sN and dN as the selected path, which incurs the minimum number of hops during relaying/forwarding of data from sN to dN.*

Definition 4. ***Residual Energy:*** *We define the Residual Energy (R_{Ei}) of a UAV as the amount of energy present in the UAV after the completion of a task. It is calculated by subtracting the amount of energy taken to survive the duration of a task from the current energy value of the UAV. Considering the current task completion time to be T units, given the energy depletion rate e_d, and energy of the UAV at t instant of time $E_i(t)$, $R_{Ei}(t) = E_i(t) - (T \times e_d)$.*

Definition 5. ***Energy Depletion Rate:*** *We define the energy depletion rate e_d as the rate at which the energy of a UAV decreases during standalone data transmissions such that no tasks are performed in parallel.*

5 Routing Path Selection

The selection of an optimized path between sN and dN is based on the formulation of an energy-based path score function $U_p(t)$. The score function considers the residual energies $R_{Ei}(t)$ of each UAV in a path under evaluation, the number of UAVs in the path n, and the task lists $J_i(t)$ of each UAV in the selected path. The motive of selecting a minimum score communication path between the source and destination UAVs is to maximize the collective residual energies of the UAVs in a selected path, increase the communication path lifetime and to reduce the processing overheads required to discover new paths in the event of loss of a UAV member in the selected path due to energy depletion.

Axiom 1. *The energy $E_i(t)$ of a UAV node at any instant of time t is constituted of the sum of energy required to complete assigned tasks $E_i(J)$ and the energy required to maintain the controls of the UAV $E_i(N_x)$ such that*

$$E_i(t) = E_i(J) + E(N_x) \tag{1}$$

The devised $U_p(t)$ needs to make sure that the UAVs included in the path do not die out during an ongoing transmission. We formulate the routing path selection score function for a path $p = \{1, 2, 3, ..., n\}$ as a function of $R_{Ei}(t)$, the number of UAVs in the path n, and $J_i(t)$. As we intend to maximize the residual energy, the score function is kept directly proportional to it. However, an increase in n or $J_i(t)$ of the path tends to reduce $R_{Ei}(t)$, and hence, is formulated to be inversely proportional to the score function devised. It is to be noted that the different paths have different values of constituent UAVs at different instants of time, which results in different values of n over time. Accommodating a steady increase in all three of these parameters results in the minimization of $U_p(t)$.

Axiom 2. *The residual energy of the system $R_{Ei}(t)$ at the t^{th} instant of time, under regular mode of operation and no means of energy replenishment, is greater than the residual energy of the system $R_{Ei}(k)$ at the k^{th} instant of time, whenever $t < k$ due to equation 1. This relation is denoted as*

$$R_{Ei}(t) - R_{Ei}(k) > 0, \quad \forall\ k - t > 0 \tag{2}$$

The formulated score function for a path p ensures maximum residual energy for the selected path, which in turn implies maximum reusability of the UAVs in that path for multiple tasks. The score function is formulated as,

$$U_p(t) = \begin{cases} \frac{\sum_{i=1}^{n} R_{Ei}(t)}{n^2 \times (1 + \sum_{i=1}^{n} J_i(t))}, & \forall i, \quad E_i(t) > (T \times e_d) \\ 0, & \exists i, \quad E_i(t) \leq (T \times e_d) \end{cases} \tag{3}$$

Equation 3 is used to formulate our objective for the optimization problem as:

$$\begin{aligned} \underset{t}{\text{Minimize}} \quad & \mathrm{U_p}(t) \\ \text{subject to} \quad & E_i(t) > (T \times e_d) \\ & e_d > 0, T > 0, n > 0 \\ & J_i(t) \in \mathbb{R}^+ \end{aligned} \tag{4}$$

Subsequently in Theorems 1 and 2, we prove that $U_p(t)$ is convex and differentiable, respectively. The convexity of the score function insinuates that the local minima attained by the function is the global minima for the scenario, whereas, differentiability in addition to the convexity of the function implies that the application of gradient descent will converge the function to a global minimum, thereby resulting in an optimized solution.

Theorem 1. *The score function $U_p(t)$ is convex.* □

Proof. A function $f : X \Rightarrow \mathbb{R}$, where X is a convex set, is convex if $\forall\ x, y \in X$ and for $0 \leq \lambda \leq 1$

$$\lambda f(x) + (1-\lambda) f(y) \geq f(u) \tag{5}$$

where $min(u) \in N(z)$ such that $N(z) = \{u \in X : ||u - z|| < 1\}$, $z = \lambda x + (1-\lambda)y$ and $||u|| = max_{1 \leq i \leq n}\{|u_i|\}$. We prove this relation for a given path $p = \{1, 2, ..., n\}$.

For establishing a point as a local minima, let $t_1, t_2 \in T$ be two temporally separate values at the extreme ends of the convex curve, and a point $t_3 \in T$ lies between t_1 and t_2 such that $t_1 \leq t_3 \leq t_2$. We can assume that $t_1 < t_2$ such that $t_3 = \lambda t_1 + (1-\lambda) t_2$ (from equation 5). For $1 \leq i \leq n$, the score function for t_1, t_2 and t_3 is represented for $m = 1, 2, 3$ as t_m such that

$$U_p(t_m) = \begin{cases} \frac{\sum_{i=1}^{n} R_{Ei}(t_m)}{n^2 \times (1 + \sum_{i=1}^{n} J_i(t_m))}, & \forall i\ E_i(t_m) > T \times e_d \\ 0, & \exists i\ E_i(t_m) \leq T \times e_d \end{cases} \tag{6}$$

For any t_j if any point j, where $R_{Ei}(t_j) \leq 0 \Rightarrow U_p(t_j) = 0$, then $\forall m \in T, m \geq j$,

$$R_{Ei}(t_m) = 0 \Rightarrow U_p(t_m) = 0 \tag{7}$$

The relation in equation 7 signifies that if the path is unable to maintain data transfer for the task at any of the points t_1, t_2, t_3, the equation 5 is proved trivially as their utilities become zero. Subsequently, applying Definition 4 to equation 5 for $t_1 \leq t_3 \leq t_2$

$$\lambda\Big(\sum_{1}^{n}(E_i(t_1) - (T \times e_d)) - \sum_{1}^{n}(E_i(t_2) - (T \times e_d))\Big) + \sum_{1}^{n}(E_i(t_2) - (T \times e_d)) \geq \sum_{1}^{n}(E_i(t_3) - (T \times e_d)) \tag{8}$$

The 1^{st} term on the left is > 0 as $0 \leq \lambda \leq 1$, and the 2^{nd} term is always greater than the right hand side of the equation (infering from Axiom 1). This leads to fact that total residual energies at time t_1 will always be $\geq$ than the total residual energies at any point of time after t_1, which satisfies the relation in equation 5.

Similarly, for sum of residual energies $TE(t)$ and the cumulative task list $TJ(t)$ along a path, equation 5 can be represented with respect to equation 3 as:

$$\frac{TE_i(t_2)}{n^2 \times (1 + TJ_i(t_2))} + \lambda\Big(\frac{TE_i(t_1)}{n^2 \times (1 + TJ_i(t_1))} - \frac{TE_i(t_2)}{n^2 \times (1 + TJ_i(t_2))}\Big) \geq \frac{TE_i(t_3)}{n^2 \times (1 + TJ_i(t_3))} \tag{9}$$

This satisfies the relation in equation 5. Additionally, it is seen that in equation 9 the decrease in $TE(t)$ on the right hand side is far more compared to the decrease in the overall $TJ(t)$, as with completion of one task the job list decrements by a single unit, whereas the energy decreases by multiple units (as inferred from Axiom 1) such that the change $\Delta R_{Ei} > \Delta JL_i$.

Hence, under all the conditions, for a given t_1, t_2, and λ the convexity of $U_p(t)$ is satisfied, which implies that any local minima of this function is the global minima of the system. □

The score value of the paths retrieved takes up values across the positive number line, and with respect to time, we prove that it is convex. This convexity insinuates that a path has only one score value, which, when reached, does not change unless external interference occurs. The formulation shows the same minimum value at 0. This implies that once any UAV is down in the path until the UAVs are called back to recharge their batteries, the score values remain the same, which happens only when the minima is 0 and with no other score value. The convexity supports that the score value remains constant at a point irrespective of the time step change, which can happen only at the local minima (which is also the global minima).

Theorem 2. *The score function $U_p(t)$ is differentiable $\forall\ t \in \mathbb{R}^+$.* □

Proof. The formulated score is a function of t, and universally $t \geq 0$. Therefore, in order to prove that U is differentiable at all $t \in \mathbb{R}^+$, we show that U' exists at all $t \in \mathbb{R}^+$. U is differentiable at t if for an infinitesimally small interval h, $\lim_{h \to 0^+}(U_p(t+h) - U_p(t))h^{-1}$ exists. Using equation 3,

$$\lim_{h \to 0^+} \frac{U_p(t+h) - U_p(t)}{h} = \frac{1}{n^2(1 + \sum_{i=1}^{n} J_i(t))} \lim_{h \to 0^+} \frac{R_{Ei}(t+h) - R_{Ei}(t)}{h} \tag{10}$$

It is to be noted that as $h \to 0^+$, the small change in residual energy is negligible such that $R_{Ei}(h) \to 0^+$. Therefore we can represent this relation as $R_{Ei}(h) = h,\ \forall\ h \to 0^+$. Additionally, $R_{Ei}(t+h)$ can be denoted as $R_{Ei}(t) + R_{Ei}(h)$. Equation 10 can be rewritten as,

$$U'(t) = \lim_{h \to 0^+} \frac{U_p(t+h) - U_p(t)}{h} = \frac{1}{n^2(1 + \sum_{i=1}^{n} J_i(t))} \tag{11}$$

With reference to equation 11, we can state that the score function U is differentiable at all $t \in \mathbb{R}^+$. □

Hence, under all the conditions, for a given t_1, t_2, and λ the convexity of $U_p(t)$ is satisfied, which implies that any local minima of this function is the global minima of the

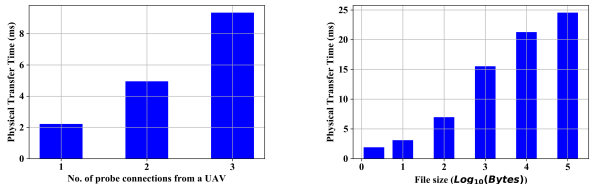
system. This leads us to believe that the paths selected based on $U_p(t)$ will be the one conserving most of the residual energies during task offload between sN and dN.

(a) Schematic outline (b) UAVs in flight

Figure 4: The hardware schematic of each of the UAVs in the tested UAV network and a small-scale networked UAV flight test for aquiring hardware metrics for tuning the simulation.

6 Hardware Evaluation and Emulation Setup

We deploy a small-scale autonomous networked UAV swarm for evaluating the real-life performance of the aerial system. The UAVs used in our work are commercially available Ardupilot-based quadrotors, which are modified to be controlled by a Raspberry Pi module. The Ardupilot module is the primary flight controller of the UAV and is immediately responsible for integrating the external and internal directional and guidance sensors to the UAV. The primary sensors integrating with the primary processor are a GPS module, direction and orientation sensors, and motor-control units. The Raspberry Pi module acts as a secondary flight-control processor and provides intelligence to the Ardupilot-based primary controller, enabling it to execute a range of flight maneuvers autonomously without any human control. The secondary controller is also responsible for establishing and maintaining communication with other UAVs in its neighborhood through its onboard Wi-Fi. The low-power Wi-Fi links formed restricts the distances for our implementation. However, it can be easily scaled-up as the metrics for large-scale simulation are considered to be relative to time and area. Fig. 4(a) shows the control schematic of each UAV in our deployment.



(a) Connection probe (b) Data transfer

Figure 5: Actual delays incurred in the UAV communication hardware during connection probe and data transfer.

We employ three such UAVs for estimating the basic performance of the communication links within the formed swarm, as shown in Fig. 4(b). Once the small-scale communication link properties – data transfer time and network connection probe time – are estimated, we include these values in an emulator along with the energy consumed by the secondary processor to emulate a large-scale network behavior. We designed this emulator using Python, which is capable of emulating large-scale UAV swarm networking scenarios based on the small-scale, real-life metrics from the actual UAVs.

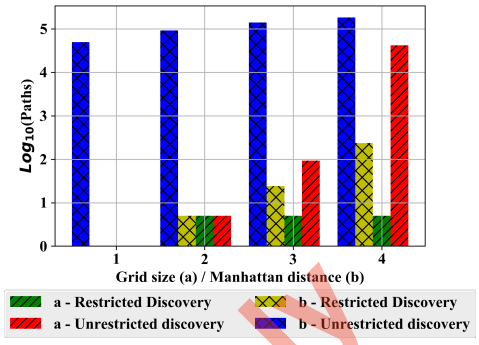


Figure 6: Comparison of the two path discovery approaches (Unrestricted discovery and Restricted discovery) based on – a) Grid size, and b) Manhattan distance.

7 Performance Analysis

This section analyzes the performance of our proposed approach, and we divide it into three parts – 1) hardware analysis of UAV links, 2) routing path discovery performance, and 3) routing path selection performance. The first part analyzes the physical UAV network link behavior against the three parameters of energy consumed, network probe time, and data transfer time. We utilize the actual values collected from the UAVs for emulating large-scale deployments in order to evaluate the performance of our proposed schemes for directional path discovery and minimum score-based routing path selection.

7.1 Hardware Analysis of UAV Links

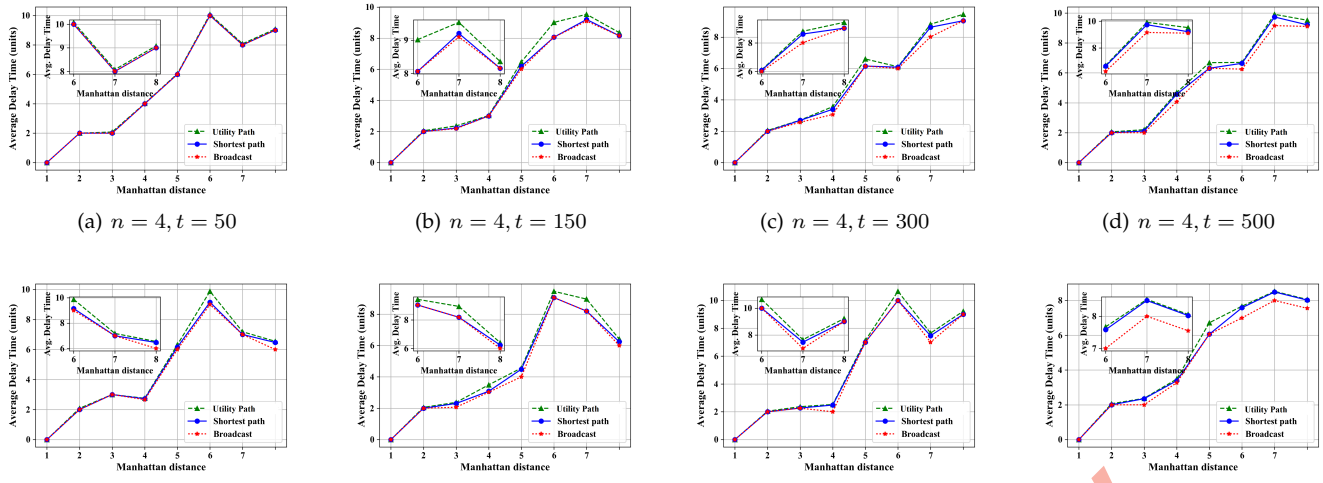
The following metrics – energy consumed, connection probe time, data transfer time – are collected from the UAV-based platform for incorporating into our large-scale emulation of the behavior of the proposed minimum score-based path selection scheme, as shown in Fig. 5.

7.1.1 Energy Consumed

Real-life metrics are collected from the UAV-based platform to incorporate them into the emulation to have similar behavior with the real-time UAVs. We observe that the UAV-based platform used 2.5 W of power under idle conditions and 3.8 W under conditions of load. We incorporate these metrics into the emulations for the energy consumed under idle conditions when a UAV is not transmitting, and under load when a UAV is transmitting to its neighboring UAV.

7.1.2 Time Taken to Probe Connections

The time taken to probe information about neighbors is measured and summarized in Fig. 5(a). The probe operation sends a request packet to each of the neighbors in the probing UAV's one-hop distance, and the probed UAV responds by sending back its information to the probing



(a) $n = 4, t = 50$ (b) $n = 4, t = 150$ (c) $n = 4, t = 300$ (d) $n = 4, t = 500$

(e) $n = 4, t = 1000$ (f) $n = 4, t = 1500$ (g) $n = 5, t = 1000$ (h) $n = 6, t = 1000$

Figure 7: A comparison of average delay times for the three approaches – Utility path, Shortest path, and Broadcast – against the Manhattan distance covered within the search area of 4×4, 5×5 and 6×6 grids.

UAV. The information consists of node identifiers, current energy levels, and tasks being currently handled by the UAV. Fig. 5(a) represents the time incurred to probe the neighbors in a one-to-many manner. The number of probed UAV platforms are varied from 1 to 3 for the three UAVs present. We observe that with an increasing number of neighbors being probed by a UAV, the average time taken to acquire its information increases, implying the presence of significant delays with an increasing number of connections.

7.1.3 Time Taken to Transfer Data

Similar to the probing operation, data packets are sent to neighboring UAVs to estimate the time taken to transfer data of varying sizes between the two connected UAVs. The data transfer metric between the neighboring UAVs is calculated in a one-to-one manner for varying data sizes. Fig. 5(b) shows the time taken by a UAV secondary controller to transfer data to its selected next-hop during the routing path selection stage of our approach. We see that with increasing data size (in bytes), the transfer time increases prohibitively. We checked the transfer time over Wi-Fi between two UAVs by having varying file sizes – 1 byte, 10 byte, 100 byte, 1 Kbyte, 10 Kbyte, 100 Kbyte, and 1 Mbyte – as shown in Fig. 5(b).

7.2 Routing Path Discovery Performance

Emulations are undertaken on our designed large-scale UAV network emulator for unrestricted and restricted path discovery algorithms as outlined in sections 4 and 4.1, respectively.

Fig. 6 shows the variations in the number of paths between a fixed sN and a selected dN. The routes are estimated based on the emulations run for varying search area grids for both restricted (directional) and unrestricted path discovery algorithms. In continuation, Fig. 6 also highlights the response of each of the two path discovery algorithms to the increase in Manhattan distance (denoted by prefix **b** in Fig. 6) between the source and destination UAVs for a fixed grid size of 4×4. The Manhattan distance (**b**) only takes into account the routes, which are along the edges of the grids, whereas the metric using the grid size (**a**) makes it possible for the paths to have diagonal paths between UAVs. As the number of diagonal paths for a $N \times N$ grid (**a**) is much smaller than the arrangement where data transmission is not allowed diagonally – Manhattan distance (**b**), the number of possible paths for **b** is much higher than that of **a**. This is true even for the restricted path discovery approach, which generates a significantly lower number of paths.

7.2.1 Unrestricted Path Discovery

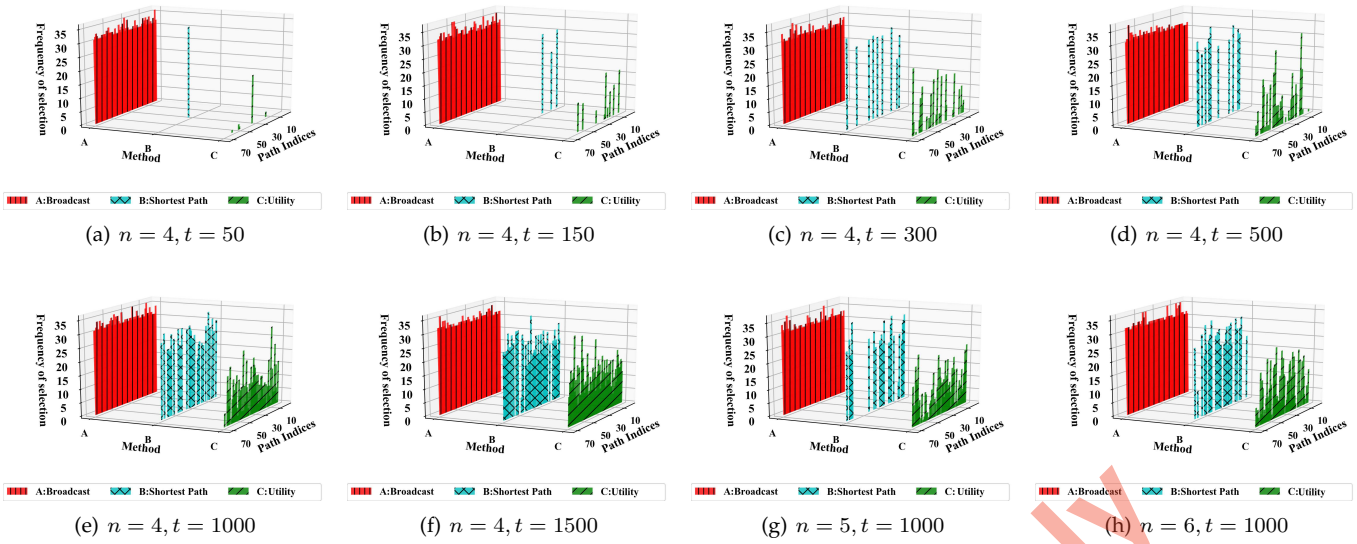
We observe that the unrestricted path discovery algorithm, which is similar to flooding, produces paths whose number increases prohibitively with the increase in the number of grids. Due to no restrictions in the direction of path search, the unrestricted version of the path discovery algorithm tends to include all the UAVs in the grid even if sN and dN are adjacent to each other, which leads to an excessive increase in overheads in most of the cases as shown in Fig. 6.

7.2.2 Directional Path Discovery

In contrast to unrestricted path discovery, the restricted/directional path discovery estimates the shortest paths between the source and destination UAVs by considering UAVs in the direction of the destination, when moving from the source UAV. The paths moving away from the destination UAVs are not considered, resulting in a significantly lesser count of paths generated. The number of paths generated by the directional path discovery algorithm remains constant irrespective of the rise in the number of grids in the search area (as denoted by prefix *a* in Fig. 6).

7.3 Routing Path Selection Performance

As outlined in the previous sections, there are three possible means of discovering the path between the source and the destination UAVs – a simple broadcast-based scheme, a



(a) $n = 4, t = 50$ (b) $n = 4, t = 150$ (c) $n = 4, t = 300$ (d) $n = 4, t = 500$

(e) $n = 4, t = 1000$ (f) $n = 4, t = 1500$ (g) $n = 5, t = 1000$ (h) $n = 6, t = 1000$

Figure 8: Comparison of the selected path and the choice of path for the three approaches with varying intervals of time in a search area of 4×4, 5×5 and 6×6 grids.

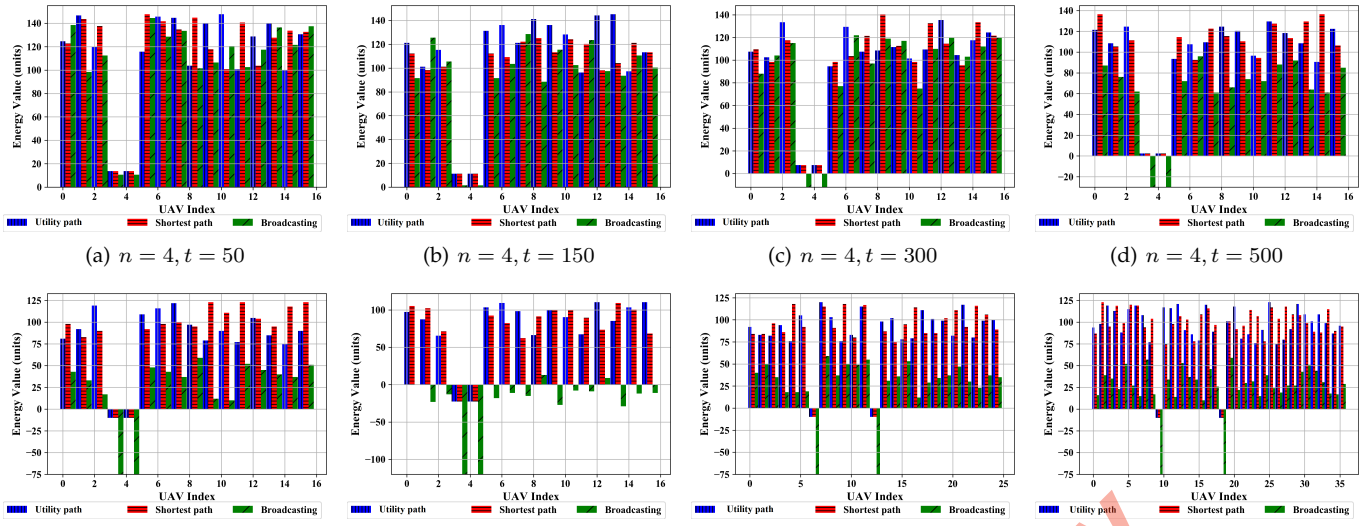
shortest path-based scheme, and our proposed minimum score-based selected path. Out of these three, the broadcast-based scheme stands out to be the one generating excessive network and processing overheads as it follows a network flooding-based approach. On account of immediate neighbor-based forwarding of packets by the nodes, the broadcast-based scheme also induces massive redundancies and energy wastage. In contrast, the shortest path-based approach produces much lesser overheads and consumes comparatively lesser energy than the broadcast-based one. However, the shortest path scheme fails to take into account the energy status of the paths being formed and the individual lifetime of the constituent UAVs. Thirdly, the paths received from the restricted path discovery algorithm are taken up for routing path selection by the formulated score function in Section 5. The path with the minimum score value is chosen as the optimal path for communication between the source and destination UAVs. The performance of routing path selection is evaluated for the three mentioned approaches through the following five metrics – average delays, the frequency of UAV selection, energy consumed by the individual UAVs, net energy consumed by each approach, and finally, number of UAVs alive after completion of a task. The minimum score approach is represented as the Utility path in the figures.

7.3.1 Average Delays Incurred

Fig. 7 compares the performance of the three schemes concerning the average delays incurred for an increase in Manhattan distance between sN and dN. Considering the edge of a square of side a, where the UAV nodes are at the corners, the Pythagorean distance will be $\sqrt{a}$. In a $N \times N$ grid, there will be $2N(N+1)$ edge paths and N^2 diagonal paths. As there are more edge paths than the diagonal paths, we chose Manhattan distance as the metric to calculate the paths (which we consider as the worst-case scenario). For the sake of evaluation of this metric, we omit the paths with communication between UAV nodes located at locations (i, j) and $(i+1, j+1)$. In Fig. 8, we only choose those sets of paths, which traverse only along the edges (and not across the diagonal) to highlight better the efficacy of our approach to the shortest path and the broadcast one (which also follow traversal along the edges only). The average delays are measured in relative units of time. Figs. 7(a) - 7(f) show the delays incurred for a 4×4 deployment zone, with a total of 16 UAVs, one in each grid location. It is seen in Fig. 7(a) that for a net operational time of 50 units, the three approaches – broadcast, shortest path, and minimum score (denoted as utility path) – incur identical delays. As we gradually increase the operational time to 150, 300, 500, 1000, and 1500 timesteps in Figs. 7(b), 7(c), 7(d), 7(e), and 7(f) respectively, we observe that the delays incurred by the shortest path and the broadcast-based approach are almost identical, whereas, the proposed minimum score (utility path)-based approach incurs slightly higher delays. This increase in delay for the proposed scheme is attributed to the change in paths selections during operation time in order to maximize the score of the formed network. The other two approaches have no such provision and hence incur relatively minor delays. Similarly, increasing the grid size from 4 to 5 and 6 in Figs. 7(g) and 7(h), respectively, shows that with the increasing number of UAVs deployed, the delays incurred by the shortest path approach becomes identical to the proposed minimum score one (utility path), whereas the broadcast-based scheme still incurs the least delays in data transfer.

7.3.2 Frequency of Path Selection

Fig. 8 outlines the results of the analysis of the three approaches for the various paths generated between sN and dN. As the number of paths generated is in tens of thousands, we represent the efficacy of our proposed approach against the two other schemes (broadcast and shortest path) for generated path indices from 1 to 70 only. The third axis



(a) $n = 4, t = 50$ (b) $n = 4, t = 150$ (c) $n = 4, t = 300$ (d) $n = 4, t = 500$

(e) $n = 4, t = 1000$ (f) $n = 4, t = 1500$ (g) $n = 5, t = 1000$ (h) $n = 6, t = 1000$

Figure 9: Comparison of energy utilized by the individual UAVs in deployments with 4×4, 5×5 and 6×6 grid sizes.

of the plots in Figs. 8(a) - 8(h) represents the frequency of selection of the generated paths during execution of the assigned tasks for varying timesteps. The $3D$ plots show frequencies of the paths selected for timesteps of 50, 150, 300, 500, 1000, and 1500 in Figs. 8(a) - 8(f) respectively for a 4×4 deployment. From these figures, we observe that the broadcast-based scheme overclocks the UAVs in all the paths resulting in a higher density of selections, as is reflected in the plots. This broadcast-based scheme transmits the data via all available paths, assuming it reaches via one or more than one of these paths to dN. In contrast, the shortest path based approach utilizes only a few selected paths, and in fact, overclocks them till the path is no more alive and sustainable due to exhaustion of energy of one or more of its constituent UAVs. Finally, our minimum score path (denoted as utility) chooses more uniformly and tries to keep as many paths alive as possible by maximizing the score of the paths selected, as is seen by the smaller frequencies of selections over multiple path indices. We observe the same trend for an increasing number of UAVs in Figs. 8(g) and 8(h) for grid sizes of 5×5 and 6×6 respectively.

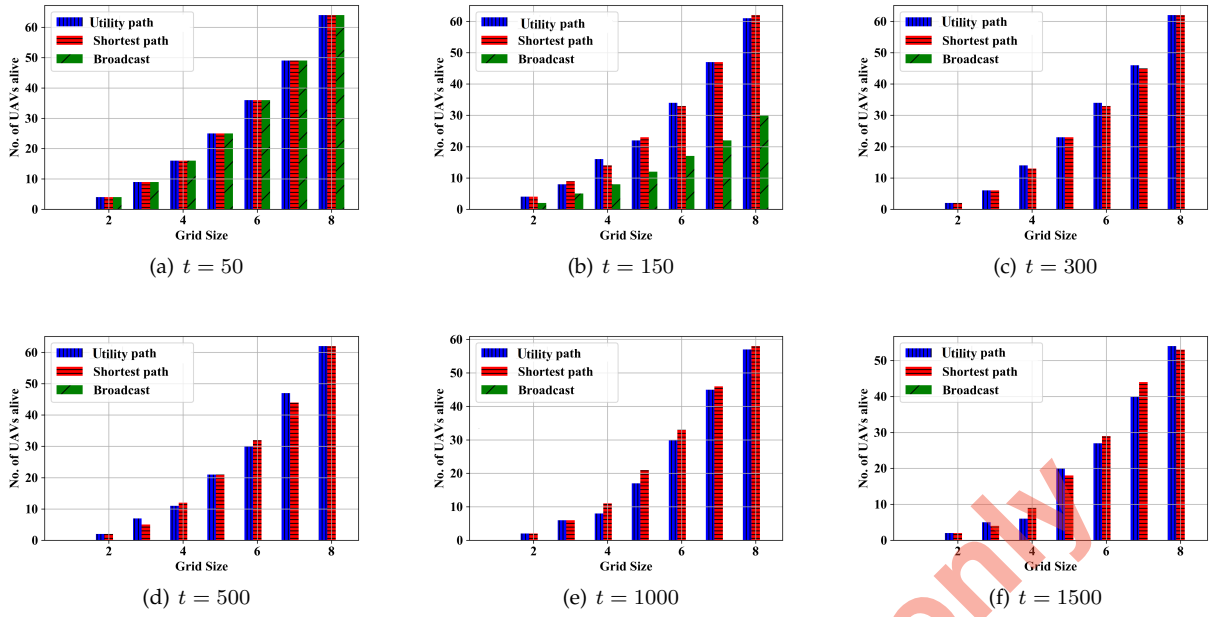
7.3.3 UAV Energies Consumed

The plots in Fig. 9 show the energy levels of all the UAVs included in the paths for the broadcast-based scheme, the proposed minimum score (utility path)-based scheme, and the shortest path selection scheme. We observe from the energy plots in Figs. 9(a) - 9(f) for a 4×4 implementation that, with increasing timesteps, the proposed minimum score (utility path)-based scheme consumes significantly lesser energies as compared to the shortest path and broadcast-based schemes. For timesteps between 300 - 500, the UAVs 4 - 6 in the broadcast-based scheme deplete all their energies (Fig. 9(c) - 9(d)) due to UAV overuse. The same set of UAVs deplete all of their energies for timesteps of 1000 (Fig. 9(e)) for all three schemes. However, for the remaining UAVs, the proposed scheme still retains maximum energies. In Fig. 9(f) the broadcast-based scheme runs out of energy for all 16 UAVs for an operational timestep of 1500. We observe the same trend in Figs. 9(g) and 9(h) for implementation sizes of 5×5 and 6×6 respectively. To sum it up, the proposed minimum score (utility path)-based scheme successfully conserves the energy of the maximum number of UAVs and performs better than the shortest path and broadcast-based schemes, making our approach more sustainable. This behavior is attributed to the repeated use of the shortest path in the shortest path algorithm based path selection, resulting in the energies of the UAVs present in the pathways to deplete faster than the remaining UAVs, not in the selected routes.

In contrast, for the minimum score (utility path)-based method, once a path is used, due to the decrease in the collective energies of UAVs in that path, the collective residual energy ($TE(t)$) of the path decreases. Our scheme automatically switches and chooses the next best path with an estimated maximum $TE(t)$. The negative energy values shown in the plots signify that the UAVs try to send data to the UAVs (the ones with negative energy) even though they are down.

7.3.4 UAVs Alive

Fig. 10 shows the plot of the number of UAVs alive post completion of the task for the three chosen approaches with varying timesteps. Figs. 10(a) - 10(f) show the number of UAVs alive for grid sizes varying from 2×2 to 8×8 for 50, 150, 300, 500, 1000, and 1500 timesteps respectively. The broadcast-based scheme sees a rapid drop in the number of UAVs alive after task completion with an increase in time steps. The broadcast scheme has UAVs alive for timesteps till 150, as shown in Figs. 10(a) and 10(b). Beyond 150 timesteps, the broadcast scheme overutilizes the UAVs such that no UAV remains alive from 500 to 1500 for any of the grid sizes, as seen in Figs. 10(c) - 10(f). The proposed minimum score-based scheme (utility path) outperforms the shortest path based one in terms of the number of UAVs alive for almost all grid sizes, as is seen in Figs. 10(a) - 10(f). This result additionally highlights the superiority of



(a) $t = 50$ (b) $t = 150$ (c) $t = 300$

(d) $t = 500$ (e) $t = 1000$ (f) $t = 1500$

Figure 10: Comparison of UAV functional after task competion at the end of designated timeframes for all three approaches.

our proposed scheme when compared to the shortest path approach.

7.3.5 *Collective Energy Consumption*

The results in Sections 7.3.2 and 7.3.4 show that the broadcast scheme is not suitable for use in optimized task distribution in our scenario. In this Section, we only consider the performance of our proposed approach against the shortest path approach. Fig. 11 shows the comparison between variations in the number of UAVs and the cumulative percentage of energy utilized by the system. The energy used by each approach is represented for the percentage of the allowable energy consumed by the member UAVs of varying grid numbers. From Fig. 11, we observe that the shortest path-based scheme frequently consumes energy, which is much higher than the allowable limit (100%). This result signifies that the permanent depletion of energies of one or more member UAVs in the swarm. Our proposed approach avoids such a scenario by continuously changing the offload task load between various UAVs in the swarm. The paths chosen by our proposed minimum score-based scheme shifts the communication load from one UAV to another to avoid path destruction due to the exhaustion of energy in one or more member UAVs of that path. Our proposed scheme also ensures distribution of load between eligible UAVs for longer hours of operation, which in turn prolongs the lifetime of the UAVs as well as the offload paths.

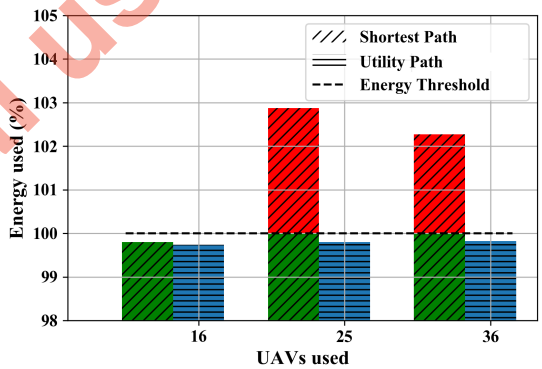


Figure 11: Comparison of cumulative energy consumed by the system with varying UAVs for the two approaches – 1) shortest path, and 2) minimum score (utility) path.

8 Conclusion

The collaborative nature of the UAV swarm considered in this work necessitates energy-efficient and low-overhead incurring solutions to accommodate the resource-constrained environment of the UAVs forming the UAV swarm and ensure the sustainability of the formed network. The proposed residual energy maximizing score minimizing-based routing path selection scheme ensures balanced energy utilization between members of the ad-hoc UAV swarm, and enhances the overall path lifetime, without incurring additional delays in doing so. This scheme in conjunction with a directional path discovery scheme allows for the reduction of overheads during unrestricted path discovery, and the delays caused thereof ensuring higher longevity of the aerial paths selected and higher reusability of the UAVs in the path when compared against established schemes such as the shortest path and broadcast-based routing. The ECoR scheme causes different UAVs to be chosen at various time instants, even between the same source and destination UAVs. This leads to some UAVs being used for additional tasks (offloading) besides their regular ones, in turn, causing different residual energies at various homogenous UAV setups. The average transmission delays in our scheme, although marginally higher than the shortest path-based

and broadcast-based schemes for smaller deployments (up to 4×4), becomes comparable to these schemes for larger deployments.

As this work insinuates the repeated polling of UAVs for their status, in the future, we plan to implement improvements to this approach by incorporating self-improving and fast online learning schemes.

REFERENCES

[1] M. Harounabadi and A. Mitschele-Thiel, "Trajectory and Buffer Aware Message Forwarding for Multiple Cooperating UAVs in Message Ferry Networks," in *Ad Hoc Networks*. Springer, 2018, pp. 156–165.

[2] G. Xie, Y. Chen, X. Xiao, C. Xu, R. Li, and K. Li, "Energy-efficient fault-tolerant scheduling of reliable parallel applications on heterogeneous distributed embedded systems," *IEEE Transactions on Sustainable Computing*, vol. 3, no. 3, pp. 167–181, 2018.

[3] Q. Gao, J. Zhang, and S. V. Hanly, "Cross-layer rate control in wireless networks with lossy links: leaky-pipe flow, effective network utility maximization and hop-by-hop algorithms," *IEEE Transactions on wireless communications*, vol. 8, no. 6, 2009.

[4] T. Habara, K. Mizutani, and H. Harada, "A load balancing algorithm for layer 2 routing based wi-sun systems," in *2017 IEEE 85^{th} Vehicular Technology Conference (VTC Spring)*. IEEE, 2017, pp. 1–5.

[5] A. Boukerche, S. Guan, and R. E. De Grande, "A Task-centric Mobile Cloud-based System to Enable Energy-aware Efficient Offloading," *IEEE Transactions on Sustainable Computing*, 2018.

[6] N. R. Lawrance, J. J. Chung, and G. A. Hollinger, "Fast Marching Adaptive Sampling," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 696–703, 2017.

[7] S.-M. Hung and S. N. Givigi, "A Q-learning approach to flocking with UAVs in a stochastic environment," *IEEE transactions on cybernetics*, vol. 47, no. 1, pp. 186–197, 2017.

[8] H. Duan, Q. Luo, Y. Shi, and G. Ma, "Hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration," *IEEE Computational intelligence magazine*, vol. 8, no. 3, pp. 16–27, 2013.

[9] P. Chandhar, D. Danev, and E. G. Larsson, "Massive MIMO for communications with drone swarms," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1604–1629, 2018.

[10] D. Wu, D. I. Arkhipov, M. Kim, C. L. Talcott, A. C. Regan, J. A. McCann, and N. Venkatasubramanian, "ADDSEN: Adaptive data processing and dissemination for drone swarms in urban sensing," *IEEE transactions on computers*, vol. 66, no. 2, pp. 183–198, 2017.

[11] D. Behnke, K. Daniel, and C. Wietfeld, "Comparison of distributed ad-hoc network planning algorithms for autonomous flying robots," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[12] S. Rao and D. Ghose, "Sliding mode control-based autopilots for leaderless consensus of unmanned aerial vehicles," *IEEE transactions on control systems technology*, vol. 22, no. 5, pp. 1964–1972, 2014.

[13] A. Bujari, C. E. Palazzi, and D. Ronzani, "A comparison of stateless position-based packet routing algorithms for FANETs," *IEEE Transactions on Mobile Computing*, vol. 17, no. 11, pp. 2468–2482, 2018.

[14] S. Rosati, K. Krużelecki, G. Heitz, D. Floreano, and B. Rimoldi, "Dynamic routing for flying ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1690–1700, 2015.

[15] Y. He, X. Tang, R. Zhang, X. Du, D. Zhou, and M. Guizani, "A Course-Aware Opportunistic Routing Protocol for FANETs," *IEEE Access*, vol. 7, pp. 144 303–144 312, 2019.

[16] H. Ghazzai, A. Feidi, H. Menouar, and M. L. Ammari, "An exploratory search strategy for data routing in flying ad hoc networks," in *2017 IEEE 28^{th} Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017, pp. 1–7.

[17] R. R. Pitre, X. R. Li, and R. Delbalzo, "UAV route planning for joint search and track missions—An information-value approach," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 3, pp. 2551–2565, 2012.

[18] C. Pu, "Jamming-resilient multipath routing protocol for flying ad hoc networks," *IEEE Access*, vol. 6, pp. 68 472–68 486, 2018.

[19] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.

Anandarup Mukherjee He is currently a Senior Research Fellow and Ph.D. Scholar in Engineering at the Department of Computer Science and Engineering at the Indian Institute of Technology, Kharagpur. He finished his M.Tech and B.Tech from West Bengal University of Technology in the years 2012 and 2010, respectively. His research interests include, but are not limited to, IoT, networked robots, unmanned aerial vehicle swarms, and enabling deep learning for these platforms for controls and communications.

Sudip Misra (M'09–SM'11) He is a Professor with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India. Prior to this, he was associated with Cornell University (USA), Yale University (USA), Nortel Networks (Canada), and the Government of Ontario (Canada). He possesses several years of experience working in academia, government, and private sectors in research, teaching, consulting, project management, architecture, software design, and product engineering roles. He has authored over 300 scholarly research papers, including 160+ journal papers. He has 11 fully authored and edited books in the areas of wireless ad hoc networks, wireless sensor networks, wireless mesh networks, communication networks and distributed systems, network reliability and fault tolerance, and information and coding theory, published by Springer, Wiley, and World Scientific. His current research interests include wireless ad hoc and sensor networks, Internet of Things (IoT), computer networks, learning systems, and algorithm design for emerging communication networks.

Vadde Santosha Pradeep Chandra He completed his integrated Bachelors and Masters in Computer Science from the Indian Institute of Technology Kharagpur in 2018. Presently he is working as a software developer at Microsoft India R & D at Hyderabad, India.

Prof. N. S. Raghuwanshi He is the Director, MANIT (Bhopal, India), and is a professor in the Department of Agricultural and Food Engineering at IIT Kharagpur. He has over 25 years of experience in research and teaching various aspects of Water Management, Information Technology Applications in Agriculture and climate change. He has made a significant contribution in the area of irrigation system performance, evapotranspiration estimation techniques, hydrological modeling of agricultural watersheds, and development of educational and design software. He has published more than 110 papers in peer-reviewed journals. He is a Fellow of the Indian National Academy of Engineering (INAE, 2016) and also Fellow of National Academy of Agriculture Science (NAAS, 2016).