SNPSG Journal of Intelligent Unmanned Systems

2026, Vol. 1, No. 1, Article 124
DOI: https://doi.org/10.65218/jius.2026.124

RESEARCH ARTICLE

Multi-UAV Opportunistic Data Collection for Hybrid Ground-Air Wireless Mesh Networks

Marcelo Paulon J.V.[1,*], Bruno José Olivieri de Souza[1] and Markus Endler[1]

1 Department of Informatics, Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rio de Janeiro, Brazil

* Corresponding author(s): Marcelo Paulon J.V.

Email(s): mvasconcelos@inf.puc-rio.br, bolivieri@inf.puc-rio.br, endler@inf.puc-rio.br

Received: 26 March 2026 | Accepted: 15 June 2026 | Published Online: 3 July 2026

Abstract: We address the problem of maximizing data collection from ground Wireless Mesh Sensor Networks (WMSNs) using UAV swarms that follow fixed trajectories. Unlike most prior works that optimize UAV routes to visit sensors, we focus on scenarios where UAVs cannot change their trajectory or speed. In a real-world scenario, this constraint could be due to the computational complexity of multi-UAV path planning (NP-hard), energy saving requirements, or operational requirements such as tight time-constraints in patrol and surveillance missions. We propose the Choreographed Max Flow algorithm, which pre-computes time-staged data routing schedules for the ground network data based on the known UAV trajectory, and NODE-GAFT, its energy-aware extension. Through 132,848 simulation runs comparing eight algorithms, we demonstrate that trajectory-aware approaches substantially outperform trajectory-agnostic approaches in fixed-trajectory scenarios. We then focus on networks with specific topological patterns where the UAV swarm encounters temporally separated contact regions. On custom 3-Shape topologies, Choreographed-MaxFlow wins 69% of paired runs against the reactive MAM baseline, with gains up to +14.0 percentage points in Data Collection Efficiency. We also propose a temporal hiatus detector (98.4% accuracy) that enables topology-adaptive algorithm selection to dynamically configure ground mesh networks when UAVs encounter them. Our results show that our proposed approaches with pre-computed routing schedules provide a significant advantage when ground networks form topologies with temporal gaps in UAV coverage.

Keywords: UAV data collection, Wireless Mesh Networks, Opportunistic data collection, Trajectory-aware data routing, Ground-air coordination, Sensor networks

I. *Introduction*

This paper investigates approaches for optimizing opportunistic data collection from ground Wireless Mesh Networks (WMNs) using swarms of Unmanned Aerial Vehicles (UAVs) that follow fixed trajectories. The collected data may include temperature and humidity readings (e.g., from low-energy sensor nodes), audio and video files (e.g., for surveillance or fauna monitoring), or large volumes of binary data from seismic sensors, enabling a wide range of real-world applications.

Wireless Mesh Networks (WMNs) and Wireless Sensor Networks (WSNs) are commonly deployed in hard-to-reach areas to facilitate data collection, as they allow nodes to forward data to a designated point of interest (PoI), such as a base station, cluster head, or mobile sink. This data-forwarding capability enables data collection from distant nodes that a data collector cannot physically reach due to time or energy constraints, thereby improving overall collection efficiency [1].

UAVs are well-suited for data collection in remote and hard-to-reach areas, such as forests and watersheds [2], where alternatives (e.g., Wide Area Network transmission or individual node surveying by land) are often impractical due to infrastructure or cost limitations. Using UAVs as mobile data collectors from WMNs can significantly improve the efficiency and speed of data retrieval, making them a compelling choice for applications such as surveillance, environmental monitoring, and disaster response [3].

Although extensive research exists on data collection from ground networks using UAVs, most of it focuses on scenarios where UAVs adapt their flight trajectories to optimize collection [3]. While such path-planning approaches can be effective, our work addresses a different constraint: a single UAV swarm travels together in formation and collects data opportunistically along a predetermined path, which we define as a fixed-trajectory constraint. Scenarios under such restrictions pose unique challenges and, as we demonstrate, offer significant opportunities for optimization through ground-air coordination.

Opportunistic UAV data collection is a subject that we found to be less frequently explored yet highly relevant for practical applications. Changing a UAV's route is expensive, both computationally (the Multi-TSP problem is NP-hard) and in terms of energy (UAV motors are the biggest battery consumers) [4]. Fixed trajectories are common in patrol and surveillance missions, periodic monitoring of infrastructure (e.g., pipelines, power lines), and other missions in which the UAV swarm must cover a predefined area systematically.

In these scenarios, the main optimization opportunity lies not in where the UAVs fly, but in how the ground network routes data toward the UAVs along their predetermined path. A ground mesh network can actively reconfigure its data routing to anticipate the UAVs' arrival at different positions, potentially collecting more data than a reactive approach that only begins routing data when UAVs are already in range.

We conducted an investigation of 128,000 simulation experiments comparing eight algorithms which reveals that, while a reactive routing approach (MAM [6]) slightly outperforms our proposed approach overall, there exist specific network topologies where pre-computed routing achieves substantially higher data collection, with up to +30.69 percentage points in individual runs, as we will present in this paper. These topologies share a common characteristic we identified: the ground sensor network forms an elongated or curved structure, and there are temporally separated contact regions for the UAV swarm's trajectory (leading to periods of time in which no data can be collected, because no sensors are within range). This observation motivated

DOI: https://doi.org/10.65218/jius.2026.124

the topology-focused experiments that are the main focus of this paper, where we investigate the relationship between network topology and algorithm performance.

A. Contributions

The main contributions of this paper are:

1. The Choreographed Max Flow algorithm: a novel algorithm that pre-computes optimal data routing schedules based on known UAV trajectories using Dinic's maximum flow algorithm, and NODE-GAFT, its energy-aware extension that scales flow network capacities by each node's projected remaining energy.

2. The concept of ground-air temporal hiatuses, which are periods where no ground nodes are within radio range of the UAV swarm, and an investigation of how these hiatuses affect algorithm performance, showing that pre-computed routing provides a significant advantage in topologies with temporally separated contact regions.

3. A temporal hiatus detector algorithm that enables topology-adaptive algorithm selection at the moment of ground-air encounter, which our experiments indicate to have 98.4% accuracy.

4. An evaluation of 132,848 simulation runs comparing eight algorithms across randomized and custom topologies, with statistical significance validated through Mann-Whitney U tests, Wilcoxon signed-rank tests, and Vargha–Delaney $\hat{A}_{12}$ effect sizes [7].

The remainder of this paper is organized as follows: Section 2 is a related work and background review. In section 3 we present our system model and proposed algorithms. In section 4 we describe our experimental setup. In section 5 we present the results of randomized experiments. In section 6 we present topology-focused experiments that are the main focus of this paper. In section 7 we discuss the findings and trade-offs we identified. Finally, we conclude in section 8 with a summary of our experiments, findings and potential directions for future work.

II. Related Work

Deploying UAVs as mobile data collectors from WSNs has become a popular approach in the last few decades, due to their flexibility and rapid deployment capabilities, being able to reach areas with difficult access. However, most existing work on this topic focuses on optimizing UAV trajectories (path planning) to visit sensor nodes, usually clustering a ground network and defining cluster heads (CHs) which the UAVs will visit. In contrast, our work shifts the focus from path planning and clusterization, to how the ground network may be reconfigured to optimize data collection given a fixed UAV swarm trajectory.

A. UAV Path Planning and Data Collection Modes

UAV path planning for data collection is an extensively explored subject, addressed through different approaches such as ant colony optimization [8, 9], Mixed-Integer Linear Programming [10], genetic algorithms [11, 12], deep reinforcement learning [13], and game theory [14]. Wei et al. [15] provide a comprehensive survey analyzing different data collection modes (hovering, flying, and hybrid). In many applications, it is unacceptable to spend energy reducing speed and waiting for full data collection, and it may also not be acceptable in time-constrained missions. Our work focuses exclusively on flying mode with fixed trajectories, where UAVs do not change their speed when collecting data.

Prior work [16] conducted both real-world and simulation experiments in flying mode under scenarios similar to ours. Their results show that UAVs can collect substantial data when passing over sensors, especially at low speeds, but also highlight a non-negligible gap between simulation and real-world results. To mitigate this concern, we focus on comparing high-level data routing strategies rather than modeling high-fidelity radio behavior, and we analyze results in terms of data collection efficiency rather than absolute data quantities.

B. Two-Phase Approaches

Many prior works on data collection from ground WSNs using UAVs divide the process into two phases: (1) assigning cluster heads in the ground network (nodes that UAVs will collect data from and that receive data from non-CH neighbors) and (2) path planning, where UAVs determine the best route to visit the CHs [3, 17, 18]. Since in all of our evaluated scenarios the UAVs fly in a pre-defined fixed route, we do not focus on the path planning problem. The aforementioned papers, however, also evaluate the cluster head assignment phase, which is relevant to our research since we want to compare classical data collection optimization approaches to ours.

There are works that consider certain formation restrictions for data collection optimization, but, to the best of our knowledge, no prior work explores the same constraints we apply: leveraging the current state of the ground mesh network and the UAVs' fixed trajectories to pre-compute optimal mesh routing schedules using maximum flow algorithms. We adapt existing CH approaches to the fixed-trajectory scenario, limiting CH candidates to nodes that are in the trajectory of the UAV swarm, enabling fairer comparisons between our algorithms and other algorithms from the literature.

C. Clustering and Energy-Efficient Approaches

Classical WSN clustering approaches include K-Means [19], which partitions sensor nodes into clusters with centroids as cluster heads, and HEED [20], which selects cluster heads based on residual energy and communication cost. LEACH [21] and its successors [22] focus on extending network lifetime through energy-balanced clustering. However, these approaches are designed for scenarios where mobile sinks visit the selected cluster heads, which is the two-phase approach (CH selection + path planning) we mentioned. In our fixed-trajectory scenario, cluster heads may fall outside the UAV path, degrading performance. To address this, we adapted K-Means and HEED to constrain cluster heads to nodes within the UAV trajectory (we call those adaptations K-Means-FT and HEED-FT).

In the context of sensor data collection, radio communications are responsible for most of the energy draw in microcontrollers [23], so minimizing radio usage is important for network lifetime. Ground network energy efficiency optimization is not our main focus, since the biggest energy consumer of the system (UAV propulsion [4]) is not affected by the evaluated approaches in any of our experiments, as UAVs do not change their trajec-

DOI: https://doi.org/10.65218/jius.2026.124

tories. Still, we measure energy consumption as a secondary metric.

D. Opportunistic Data Collection

Among approaches for opportunistic data collection, Liu et al. [14] propose a coalition formation game for air-ground collaborative planning where ground sensors actively form coalitions to improve upload efficiency. Their approach includes the ability of UAVs changing their speed to collect more data, which we do not implement since it would violate our system model constraints. The MAM algorithm [6] uses reactive mesh routing where data flows toward the closest UAV through forwarding trees that update dynamically as UAVs move and send heartbeats. Unlike our proposed approaches, MAM reacts in real-time to UAV proximity instead of pre-computing data routing schedules.

III. System Model and Proposed Approach

A. System Model

We consider the following assumptions for our system model:

- Multiple sensor nodes form a single Mesh WSN (all ground nodes can communicate through multi-hop). Network partitioning does not occur during data collection.
- All sensor nodes and UAVs have the same radio range and bandwidth.
- UAVs travel together in formation (UAV swarm) and form a connected air network.
- UAVs have significantly higher storage capacity than sensor nodes and never reject data collection when connected to a sensor.
- Sensor nodes have accurate position information and can detect their neighbors.
- There is no radio interference and guaranteed delivery when a node sends a message to another within range.
- When a node forwards data to a neighbor, the data is removed from the sender's buffer (i.e., data forwarding is a move operation, not a copy). Each data unit exists at exactly one node at any given time.
- UAVs do not change their trajectory or speed during data collection.
- Sensor nodes and UAVs do not fail during the data collection process.

Our simple radio model assumes that there is a fixed data rate (250 data units per second), a fixed radio range (100 map units), that the TX energy cost is 2 energy units per transmission of a data unit, and that the RX energy cost is 1 energy unit per reception of a data unit (MTU is 1). We assume that all nodes have the same radio range and bandwidth, and that there is no packet fragmentation, no QoS and no collisions or retransmissions due to interference.

We acknowledge that, for real-world scenarios, several of these assumptions are not realistic. For instance, there are no reports of radio systems with zero interference; UAVs may fail or deviate from their planned trajectories due to weather conditions or GPS errors; radio range and bandwidth may vary due to environmental factors [16]. However, for our research goals, these assumptions allow us to focus on trajectory-aware optimization of data collection without modeling complex radio behavior and failure models.

A mesh network $G = (S, E)$, where S is the set of sensor nodes and E the set of edges, satisfies: (1) every connected pair is within communication range ϕ; (2) every node has at least one neighbor; and (3) any two nodes in S can communicate to each other through multi-hop paths (a list of edges in E).

We define the Coverage Deficit Ratio (CDR) as the percentage of ground nodes that are outside the UAV swarm's radio coverage along its entire trajectory. A CDR of 10% means 10% of nodes are never in direct radio range of any UAV during the swarm's visitation, while an 80% CDR represents a scenario where most nodes rely on mesh forwarding to deliver their data to the swarm. CDR is an important parameter in our experiments, as it directly affects how much data must traverse the mesh network before reaching a UAV.

Fig. 1 illustrates a data collection scenario we evaluate, showing a UAV swarm following a fixed trajectory (green dashed arrow) over a ground mesh sensor network.

B. AirSwarm-Mesh-Encounter (AMEnc) Approach

We introduce the AirSwarm-Mesh-Encounter (AMEnc) approach, a phase-based model for describing and comparing any UAV–WMN data collection strategy. AMEnc decomposes the data collection process into distinct temporal phases, which makes algorithms easier to understand, compare, and replicate, regardless of how fundamentally different their designs may be. We define the following phases, which occur in sequence during a ground-air encounter:

1. START: The UAV swarm is airborne but has not yet connected to any ground node. Algorithms that require pre-encounter computation (e.g., clustering approaches that elect cluster heads based on node positions) perform their main logic during this phase.
2. MEET: The first UAV connects to a ground node, establishing initial contact between the air and ground networks. During this phase, the two networks may exchange metadata (e.g., node positions, stored data quantities, energy levels, and the UAV trajectory), enabling ground-air coordination decisions.
3. ASSIGN: UAVs and WMN nodes negotiate assignments (which UAVs are assigned to which ground nodes) and routing decisions based on information exchanged during MEET. This may range from simple cluster head selection to computing optimized multi-stage routing schedules. Some algorithms (e.g., MAM's reactive routing) skip this phase entirely.
4. CONN: UAVs connect to assigned nodes and collect data according to the chosen routing strategy. This is the phase where actual data collection occurs. Algorithms with reactive routing (e.g., MAM) form and update their forwarding structures dynamically during this phase.
5. DISC/END: UAVs disconnect from the ground network when leaving radio range, completing the encounter.

 DOI: https://doi.org/10.65218/jius.2026.124

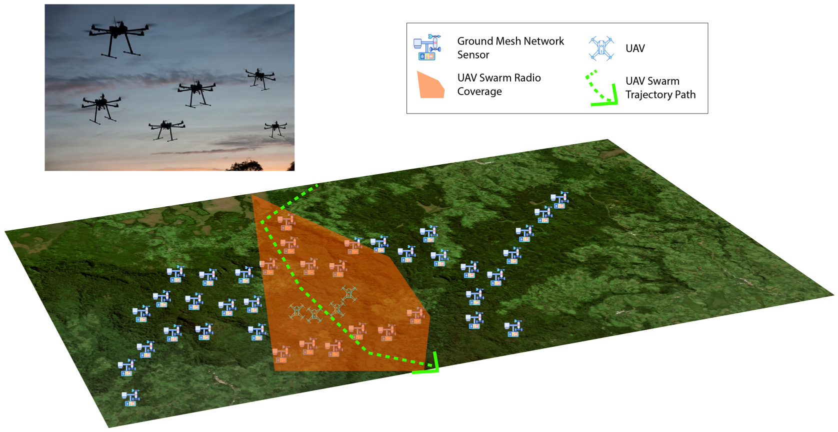


Fig. 1 Illustration of a UAV data collection scenario with ground mesh network sensor nodes forming a wireless mesh network. The orange region shows the UAV swarm's radio coverage area as it travels along the trajectory path indicated by the green arrow.

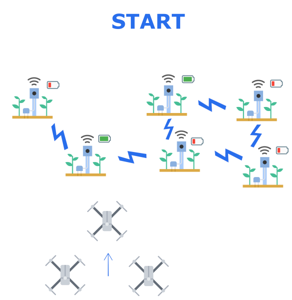
Fig. 2 illustrates three important AMEnc phases with an example scenario. In the START phase, three UAVs travel in formation toward a set of ground nodes forming a WMN; nodes have different energy levels represented by individual energy bars. In the MEET phase, the first UAV connects to a ground node, establishing the initial ground-air contact. In the CONN phase, each UAV is assigned to specific ground nodes (U1→S2, U2→S3, U3→S6) and collects data according to the algorithm's routing strategy.

By mapping each algorithm onto AMEnc phases, we can identify where its main optimization logic is implemented. For instance, clustering algorithms (K-Means, HEED) perform their main logic during START (electing cluster heads before the encounter), while our Choreographed-MaxFlow algorithm concentrates its optimization during ASSIGN (computing max-flow schedules after receiving network metadata in MEET). MAM's reactive routing has no ASSIGN phase; instead, its forwarding trees form dynamically during CONN via heartbeat propagation. We use some of those definitions as part of our algorithm descriptions and when analyzing results.

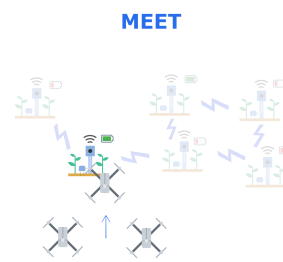
C. Evaluation Metrics

We evaluate the following metrics:

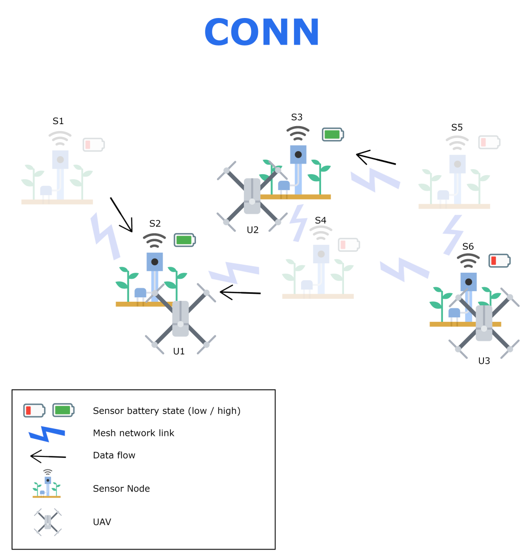
- **QDC**: Quantity of data collected by the UAV swarm (bigger is better).
- **QEC**: Quantity of energy consumed by the ground mesh network via TX+RX radio activity (smaller is better).
- **DCE**: Data Collection Efficiency = QDC / total data available (bigger is better).
- **DCE+**: DCE normalized by the Sensor Data Collection Ceiling (SDCC), which is the theoretical maximum data the UAVs could collect given radio constraints, trajectory, and bandwidth. We compute SDCC for each scenario, via simulation, assuming all sensors have infinite data available and transfer data to UAVs whenever in range, while respecting all constraints (bandwidth, radio range, connection limits). DCE+ provides fairer comparison across scenario configurations with different CDR values, UAV speeds, and maps.



(a) START phase.



(b) MEET phase.



(c) CONN phase.

Fig. 2 AMEnc phases: (a) UAV swarm approaches the ground network; (b) first ground-air contact is established; (c) UAVs connect to assigned nodes and collect data.

- **EEff**: Energy Efficiency = QDC / QEC (bigger is better). This metric penalizes algorithms that waste energy on forwarding data without successful UAV data collection, and rewards algorithms that efficiently route data toward the UAVs.

DOI: https://doi.org/10.65218/jius.2026.124

D. Evaluated Algorithms

We evaluate eight algorithms which we divide into different categories:

K-Means and **K-Means-FT**: K-Means clusters ground nodes with fixed K=6 clusters. K-Means-FT constrains cluster head selection to nodes within the UAV swarm's radio range along the trajectory, ensuring all cluster heads are visited.

HEED and **HEED-FT**: HEED [20] selects cluster heads based on residual energy and communication cost. HEED-FT applies the same fixed-trajectory adaptation as K-Means-FT.

MAM: The Monitored Arboretum Mesh algorithm [6]: uses reactive routing, making data flow toward the closest UAV through forwarding trees that are generated from recursively propagated heartbeats, constantly updated as UAVs move.

Liu 2020: Coalition formation approach [14] where ground sensors form coalitions with UAVs based on their fixed trajectory. UAVs collect data only from active coalition members.

Choreographed-MaxFlow (Ours): Pre-computes time-staged routing schedules based on the UAV trajectory using maximum flow optimization. Described in detail below.

NODE-GAFT (Ours): Energy-aware extension of Choreographed-MaxFlow that scales flow capacities by projected node energy, with the intent of making more energy-efficient routing decisions.

For fair comparison, UAVs collect data from any node within radio range (not just cluster heads), except for Liu 2020 which retains its original active-coalition restriction (as it was the only approach we selected that has fixed trajectories as part of its original design).

E. Choreographed Max Flow Algorithm

Our algorithm builds on the maximum flow problem, which is a classical optimization problem first formulated by Harris and Ross in a 1955 paper [25] which analyzed the problem in terms of a railway network with capacities. Ford and Fulkerson [26] proposed the first algorithmic solution for the problem, an year later. In our scenario, the "railway network" is the mesh sensor network, the sources are all ground nodes (connected to a virtual super-source), and the sinks are the UAVs (connected to a virtual super-sink [27]). We compute edge capacities considering bandwidth and contact duration, and use Dinic's algorithm [28] to solve the max flow. We chose this algorithm due to its efficiency and suitability to our problem size with $O(S^2E)$ time complexity [29].

The route schedule's goal is to attract as much data as possible to nodes within the UAV swarm's radio range during the entirety of their visitation. Routes for data forwarding change as UAVs traverse the ground network automatically (without the need of heartbeats or the UAVs being connected to the ground network), in a choreographed manner: each ground node receives a list of nodes it should forward data towards over time (with time deltas indicating when routing changes should occur). Unlike reactive approaches (e.g., MAM [6]) that make routing decisions on-the-fly, this algorithm predicts the UAV trajectory and computes those time-staged routing plans aiming to maximize data throughput globally over time. The data forwarding schedule is pre-computed by the UAVs during the encounter, after they receive the ground network configuration.

The algorithm divides the UAV flight path into discrete time stages, predicts which sensor nodes will be in range at each stage, builds a flow network (with super-source connecting all sensors and super-sink connecting all UAVs [27]), and solves a maximum flow problem for each stage. We set edge capacities to be the bandwidth × stage duration for node-to-node edges, and to the node's stored data capacity for source-to-node links. We then extract the routing schedule from the max-flow solution: for each node, we examine the flow allocated across all its outgoing edges (to neighbors in the mesh and to UAVs in range) and select the single neighbor with the highest allocated flow as the node's forwarding target for that stage (line 11 in Algorithm 1). This reduces the multi-commodity flow solution to a forwarding rule in which, during each time stage, each node forwards the data to exactly one next-hop neighbor. Algorithm 1 presents the pseudocode for this algorithm.

Algorithm 1 Choreographed Max Flow Algorithm

Require: UAV swarm trajectory and speed, sensor network graph $G = (S, E)$, node capacity C_n for each sensor n
1: **Constants:** bandwidth B, radio range ϕ
2: $stages \leftarrow$ compute from ϕ, UAV trajectory and speed
3: **for** each stage $\langle t_{\text{start}}, t_{\text{end}}, R_{\text{nodes}} \rangle$ in $stages$ **do**
4:     $\Delta t \leftarrow t_{\text{end}} - t_{\text{start}}$, $s \leftarrow$ super source, $z \leftarrow$ super sink
5:     Add edge (s, n) with capacity C_n for each sensor $n \in S$
6:     Add edge (n, m) with capacity $B \cdot \Delta t$ for each $(n, m) \in E$
7:     Add edge (n, u) with capacity $B \cdot \Delta t$ for each $n \in R_{\text{nodes}}$, UAV u
8:     Add edge (u, z) with capacity ∞ for each UAV u
9:     Compute max flow from s to z (Dinic's algorithm)
10:     **for** each sensor $n \in S$ **do**
11:         $f^* \leftarrow \arg\max_v \text{flow}(n, v)$
12:         Schedule n to forward data to f^* during $[t_{\text{start}}, t_{\text{end}})$
13:     **end for**
14: **end for**

The algorithm has the following properties: (1) Data-aware: edge capacities reflect stored data quantities, received upon AMEnc MEET; (2) Trajectory-aware: flow network construction uses predicted UAV positions; (3) Locally optimal: max flow is optimal within each stage but not globally across stages; (4) Polynomial: time complexity is $O(k \cdot S^2E)$; (5) Predictive: routing decisions are pre-computed before UAVs arrive to their assigned nodes.

NODE-GAFT extends this by scaling edge capacities by an energy factor $\sigma_{\min} + (1 - \sigma_{\min}) \cdot (e/e_{\max})^{\beta}$ where β=2.0 and $\sigma_{\min}$=0.05, penalizing energy-depleted relay nodes. We track projected energy cumulatively across stages. After computing each stage's max flow, we estimate TX and RX energy costs for each node and update its projected energy accordingly, so that later stages avoid overloading nodes that were heavily used in earlier stages. Algorithm 2 presents the NODE-GAFT pseudocode.

IV. Experimental Setup

We implemented a discrete-time [30] simulator in Python. The simulator updates UAV positions and network state every 10 ms. Our experiments used the following parameters:

- Number of UAVs: 5 in fixed "V-shape" formation
- Map dimensions: 800 × 600 map units

DOI: https://doi.org/10.65218/jius.2026.124

Algorithm 2 NODE-GAFT Algorithm

Require: UAV swarm trajectory and speed, sensor network graph $G = (S, E)$, node capacity C_n, initial energy $E_0[n]$ for each sensor n

Constants: bandwidth B, radio range ϕ, energy costs α_{tx}, α_{rx}, β=2.0, $\sigma_{\min}$=0.05
$E_{\text{proj}}[n] \leftarrow E_0[n]$ for all $n \in S$
$stages \leftarrow$ compute from ϕ, UAV trajectory and speed
for each stage $\langle t_{\text{start}}, t_{\text{end}}, R_{\text{nodes}} \rangle$ in $stages$ **do**
  $\Delta t \leftarrow t_{\text{end}} - t_{\text{start}}$, $s \leftarrow$ super source, $z \leftarrow$ super sink
  Add edge (s, n) with capacity $C_n \cdot \text{ESCALE}(E_{\text{proj}}[n])$ for each $n \in S$
  Add edge (n, m) with capacity $B \cdot \Delta t \cdot \text{ESCALE}(E_{\text{proj}}[n])$ for each $(n, m) \in E$
  Add edge (n, u) with capacity $B \cdot \Delta t$ for each $n \in R_{\text{nodes}}$, UAV u
  Add edge (u, z) with capacity ∞ for each UAV u
  Compute max flow from s to z (Dinic's algorithm)
  Extract routing schedule (same as Algorithm 1)
  for each node $n \in S$ **do**
    $E_{\text{proj}}[n] \leftarrow E_{\text{proj}}[n] - \alpha_{tx} \cdot \text{out}(n) - \alpha_{rx} \cdot \text{in}(n)$
  end for
end for
function ESCALE(e): **return** $\sigma_{\min} + (1 - \sigma_{\min}) \cdot (e / \max_n(E_0[n]))^{\beta}$

- Radio range: 100 map units for all nodes (ground and UAVs)
- Bandwidth: 250 data units per second
- Energy cost: 2 energy units per TX, 1 energy unit per RX
- UAV speeds: {0.02, 0.05, 0.1, 0.2, 0.4} map units/ms
- Number of ground nodes: {20, 40, 60, 80}
- Coverage Deficit Ratio (CDR): {10%, 30%, 40%, 80%}
- Data distribution: {random, 100, 250, 500} units per node
- Runs per configuration: 50 (different random seeds)

We used the same randomized map for all evaluated algorithms in each run, enabling fair comparison between them. Each configuration was repeated 50 times with different random seeds, ensuring randomized maps for each repetition. The total number of experiments is: 8 algorithms $\times$ 4 node counts $\times$ 4 CDR values $\times$ 5 speeds $\times$ 4 data distributions $\times$ 50 repetitions $=$ 128,000 simulation runs.

The simulator generates maps by randomly distributing nodes across the map area, constrained to form a fully connected mesh network and respect the target CDR parameter. Ground nodes are static and have a limited radio range of 100 map units. The UAV swarm flies from one edge of the map to the other in a straight line, collecting data opportunistically from any node within radio range. The simulation ends when the UAV swarm exits the map area. Each simulation run measures QDC, QEC, DCE, DCE+, and EEff as we defined in our metrics section.

We also ran topology-focused experiments, which we explain in next sections. For those, we first ran 240 simulations on two handcrafted maps (one per topology shape), and then generated randomized maps using a custom topology generation algorithm (Algorithm 3) with two topological patterns (3-Shape and inverted-C). The randomized topology experiments used two network sizes (60 and 80 nodes), three UAV speeds (0.05, 0.1, 0.2), three initial data distributions (100, 250, 500), and a repetition parameter of 16 (yielding 64 distinct maps), totaling 4,608 additional simulation runs. The overall total is $128{,}000 + 240 + 4{,}608 = 132{,}848$ simulation runs.

V. Overview of Randomized Experiments

We conducted 128,000 experiments (16,000 per algorithm) across randomized mesh networks. Table 1 summarizes the overall DCE results, and Fig. 3 shows the DCE distribution for each algorithm.

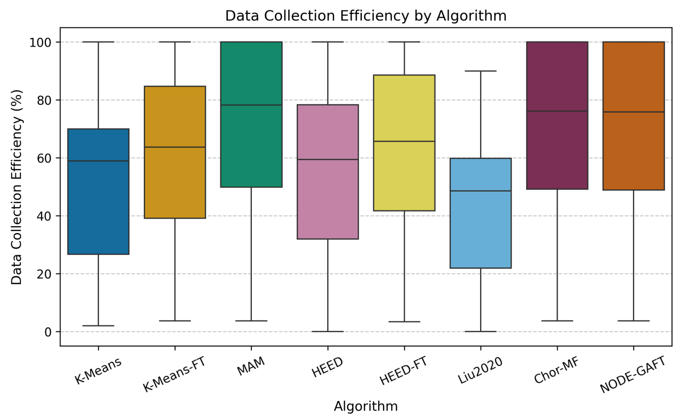


Fig. 3 DCE distribution by algorithm across all 128,000 randomized experiments. Bigger is better.

Table 1 Data Collection Efficiency (%) summary across 128,000 experiments.

Algorithm	Mean	Median	Std Dev	Min
K-Means	53.55	58.89	24.97	2.00
K-Means-FT	61.22	63.73	27.62	3.65
MAM	71.54	78.25	28.45	3.65
HEED	56.24	59.38	28.69	0.00
HEED-FT	63.21	65.71	27.96	3.38
Liu 2020	44.57	48.55	20.64	0.00
Chor-MF (Ours)	70.54	76.18	28.55	3.65
NODE-GAFT (Ours)	70.30	75.87	28.30	3.65

Our Choreographed-MaxFlow (Chor-MF) and NODE-GAFT approaches achieve competitive performance with mean DCE of 70.54% and 70.30% respectively, close to MAM's mean of 71.54%. All three trajectory-aware approaches significantly outperform trajectory-agnostic approaches: HEED (56.24%), K-Means (53.55%), and Liu 2020 (44.57%). The fixed-trajectory adaptations (K-Means-FT: 61.22%, HEED-FT: 63.21%) improve their original versions by 7–8 percentage points but are ranked lower than trajectory-aware approaches.

We observe from Table 1 that all algorithms except Liu 2020 occasionally achieve 100% DCE in favorable scenarios (e.g., low CDR, few nodes), but their minimum performance varies: HEED drops to 0.00% and Liu 2020 to 0.00%, while our approaches never fall below 3.65%.

We believe Liu 2020's low performance is due to our constrained adaptation of the original algorithm: Liu 2020 includes an online speed-planning component that allows UAVs to slow down over dense regions to collect more data, which we do not

 DOI: https://doi.org/10.65218/jius.2026.124

implement since our system model enforces fixed UAV speed. This removes an important optimization mechanism from their approach, and therefore our comparison should be interpreted as an evaluation of their coalition formation component in isolation under fixed-trajectory constraints. Additionally, UAVs only collect data from active coalition nodes, which places Liu 2020 in an unfavorable position compared to the other evaluated approaches in which UAVs may collect data from any node within their radio range. These results indicate that the Liu 2020 approach is better suited for scenarios where UAVs can adapt their speed.

A. *Scalability and CDR Analysis*

In challenging scenarios ($\geq$60 nodes, $\geq$30% CDR), optimization starts to matter more, as routing decisions in the ground mesh network become more critical for high data collection efficiency (more data has to be routed towards the nodes inside the UAV swarm's trajectory radio range). Figure 4 shows violin plots for these scenarios. MAM, Choreographed-MaxFlow and NODE-GAFT have distributions centered around higher DCE values (60–75%) with a high concentration of instances close to 100%, while K-Means, HEED, and Liu 2020 show distributions shifted toward lower DCE values, indicating less predictable performance in many scenarios.

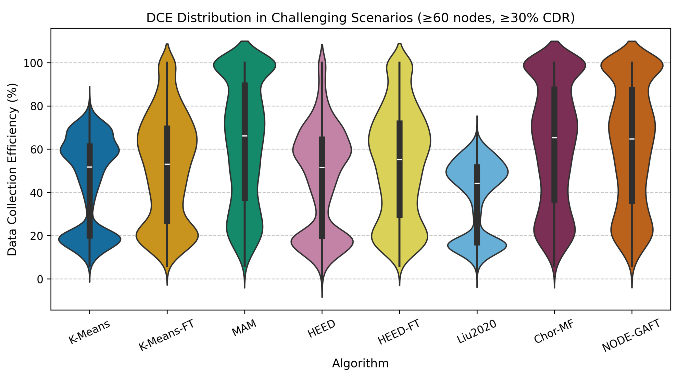


Fig. 4 DCE distribution in challenging scenarios ($\geq$60 nodes, $\geq$30% CDR).

Fig. 5 shows the relative DCE improvement of each algorithm over K-Means as baseline (median DCE 58.9%). Liu 2020 performs 17.6% worse than K-Means. HEED shows only a 0.8% improvement. The trajectory-adapted versions K-Means-FT and HEED-FT achieve 8.2% and 11.6% improvements, respectively. NODE-GAFT and Chor-MF achieve 28.8% and 29.4% improvements, while MAM achieves the highest improvement of 32.9%.

Fig. 6 shows a heatmap of median DCE by algorithm and scenario (node count and CDR). Each cell aggregates 1,000 runs (50×5 speeds $\times 4$ data distributions). We observe that a pattern repeats every 4 columns as CDR is reset for a new node count. MAM, Chor-MF and NODE-GAFT maintain green coloring even in moderately challenging scenarios (40% CDR), while Liu 2020 and K-Means show consistently red and orange cells. At the extreme (80 nodes, 80% CDR), MAM achieves 27.8% median DCE, while Chor-MF reaches 26.0%. Liu 2020 has the same DCE (15%) for all 80% CDR scenarios, indicating poor adaptation to our constraints.

Fig. 7 shows a similar heatmap using DCE+ instead of DCE. An interesting observation is that, unlike DCE, DCE+ values are higher for the most challenging scenarios (80% CDR). This

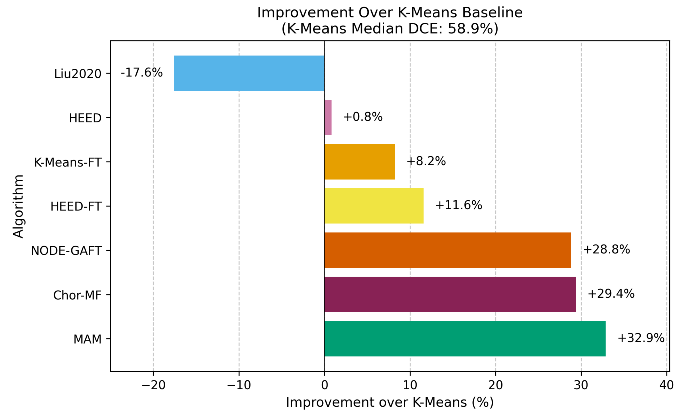


Fig. 5 Relative DCE median improvement over K-Means. Bigger is better.

inverse direction occurs because the SDCC is also lower in challenging scenarios (fewer nodes in range, less contact time), making even modest amounts of collected data a larger fraction of the data collection ceiling. Across all scenarios, MAM, Chor-MF and NODE-GAFT show significantly higher DCE+ values compared to other algorithms.

Table 2 provides a comparison of the eight evaluated algorithms, summarizing their approach, trajectory awareness, and performance characteristics we observed in our experiments.

Fig. 8 compares median DCE and DCE+ side-by-side. While the rank between algorithms is preserved, DCE+ reveals a smaller gap between our approaches and MAM (30.9% vs 29.7%/29.6%) than raw DCE medians indicate (78.25% vs 76.18%/75.87%).

B. *Head-to-Head Competitiveness with MAM*

To compare Choreographed-MaxFlow with MAM more precisely, we performed a head-to-head comparison on 16,000 common scenario pairs, with each pair using the same randomized map for both algorithms. Table 3 shows that Choreographed-MaxFlow wins (higher DCE) in 3,939 runs (24.62%), ties in 4,765 (29.78%), and loses in 7,296 (45.60%). While MAM wins more often overall, a substantial portion of runs result in ties, indicating that both algorithms perform similarly in many scenarios.

Table 4 splits the DCE outcomes by CDR. At CDR = 80%, our approach wins in 26.65% of runs (30.4% among non-tied pairs), with a mean DCE advantage of 3.13 pp and a maximum of 30.69 pp. The largest individual gains occur at high CDR, though wins are distributed across all CDR levels.

Table 5 shows the influence of UAV speed on the win rate. The highest win rate (44.56%) occurs at the fastest speed (0.4), while the largest mean advantage (4.46 pp) occurs at the slowest speed (0.02). This suggests that at slower speeds, the precomputed routing plans can achieve larger improvements in certain individual runs, but at faster speeds, more runs benefit from the optimization (with smaller individual gains).

We manually inspected the map topologies of several top-winning cases (the runs with the largest DCE differences in favor of Choreographed-MaxFlow). We observe a recurring pattern: many of these maps exhibit what we call a "C-shape" topology, meaning that the ground sensor network is arranged in a curved or elongated shape, and the UAV swarm's trajectory covers only the two extremities. In these scenarios, Choreographed-

DOI: https://doi.org/10.65218/jius.2026.124

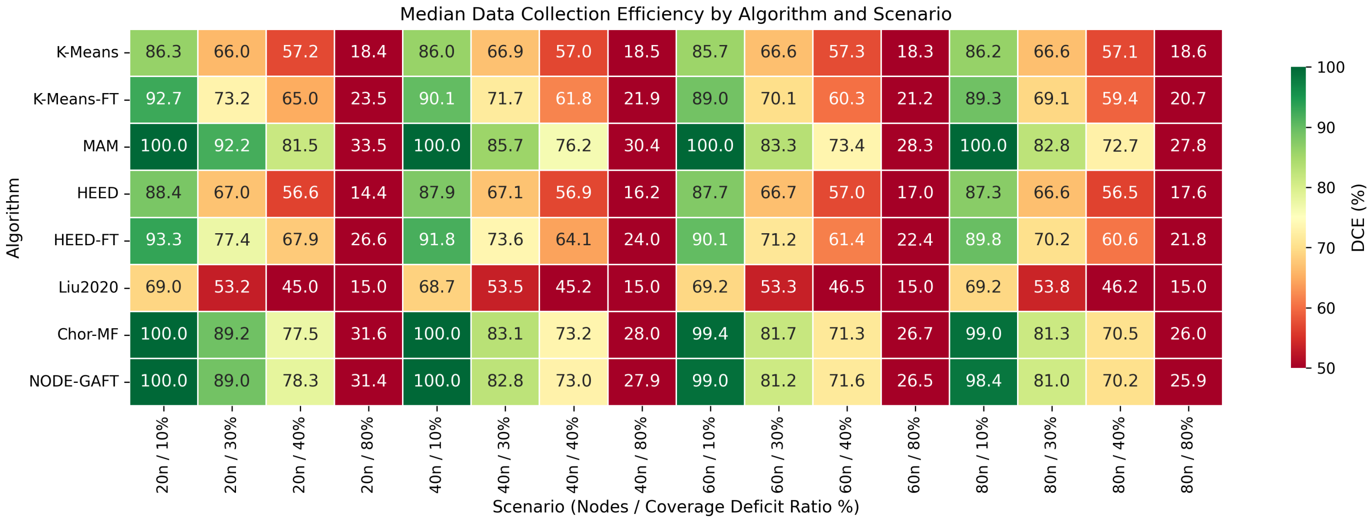


Fig. 6 Median DCE heatmap by algorithm (rows) and scenario (columns: node count / CDR). Color-coded from red (low) to green (high). Each cell = 1,000 runs. Bigger (greener) is better.

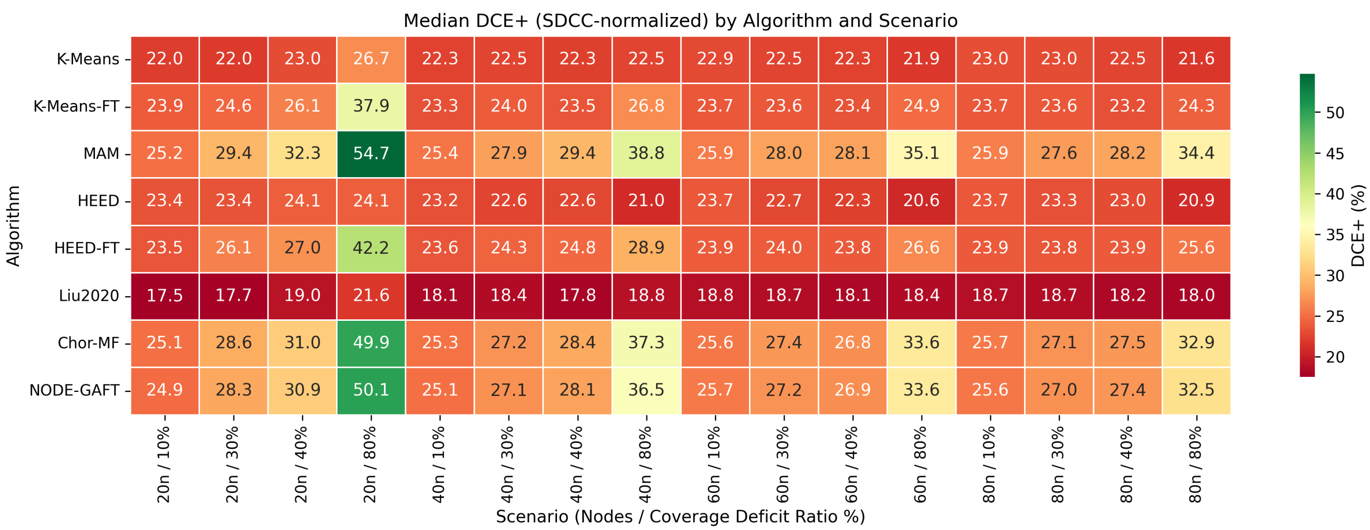


Fig. 7 Median DCE+ heatmap by algorithm and scenario. Color-coded from red to green. Bigger (greener) is better.

Table 2 Comparison of evaluated approaches for UAV-assisted data collection from mesh networks.

Algorithm	Routing Paradigm	Trajectory-Aware	Mean DCE	Median DCE	Main Characteristic
K-Means	Clustering (spatial)	No	53.55%	58.89%	Simplest clustering approach
K-Means-FT	Clustering (trajectory)	Yes	61.22%	63.73%	+8 pp over K-Means
HEED	Clustering (energy)	No	56.24%	59.38%	Energy-aware CH selection
HEED-FT	Clustering (traj.+energy)	Yes	63.21%	65.71%	+7 pp over HEED
Liu 2020	Coalition formation	Partial	44.57%	48.55%	Designed for speed adaptation
MAM	Reactive mesh routing	Yes	71.54%	78.25%	Best overall on random maps
Chor-MF (Ours)	Pre-computed max flow	Yes	70.54%	76.18%	Best on topologies w/ hiatuses
NODE-GAFT (Ours)	Energy-aware Chor-MF	Yes	70.30%	75.87%	Avoids energy depletion

 DOI: https://doi.org/10.65218/jius.2026.124

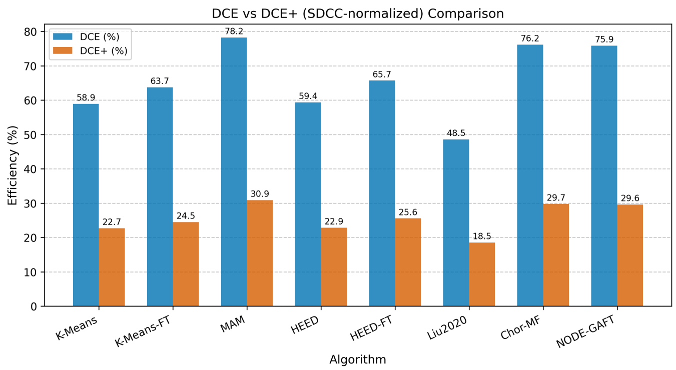


Fig. 8 Side-by-side comparison of median DCE and DCE+ by algorithm.

Table 3 Chor-MF vs MAM: Per-Run DCE outcomes across 16,000 paired experiments.

Outcome	Count	%
Chor-MF wins	3,939	24.62
Tie	4,765	29.78
MAM wins	7,296	45.60

Table 4 Chor-MF vs MAM: DCE head-to-head by CDR. Mean/Max Advantage in percentage points (pp).

CDR	Wins	Losses	Ties	Mean (pp)	Max (pp)
10%	821	1,124	2,055	+1.81	+12.70
30%	1,026	1,743	1,231	+1.79	+15.01
40%	1,026	1,993	981	+1.87	+18.12
80%	1,066	2,436	498	+3.13	+30.69

Table 5 Chor-MF vs MAM: Win rate and DCE advantage by speed.

UAV Speed	Win Rate (%)	Mean Adv. (pp)
0.02	9.38	4.46
0.05	13.19	3.16
0.10	19.97	2.92
0.20	36.41	1.96
0.40	44.56	1.23

MaxFlow's pre-computed scheduling design provides a significant advantage: when the UAV swarm first reaches the top extremity of the C shape, the algorithm computes that data should begin flowing toward the bottom in anticipation of the swarm's arrival to that region. MAM's reactive routing initially directs all data toward the top (the nearest UAV), then reverses direction only when it arrives at the other region; but the data travel time through the mesh is not fast enough to complete this redirection, resulting in less data collected overall.

C. Energy Analysis

Fig. 9 shows the trade-off between mean DCE (y-axis) and mean energy consumption (x-axis) for each algorithm. Upper-left is better (less energy consumed, higher DCE). MAM, Chor-MF, and NODE-GAFT occupy similar positions, but Chor-MF and NODE-GAFT are shifted toward lower energy consumption at comparable DCE levels. Liu 2020, K-Means, and HEED consume the least energy overall, because they collect less data.

Fig. 10 presents boxplots of the energy efficiency distribution. Liu 2020 achieves the highest EEff (median 1.14) because

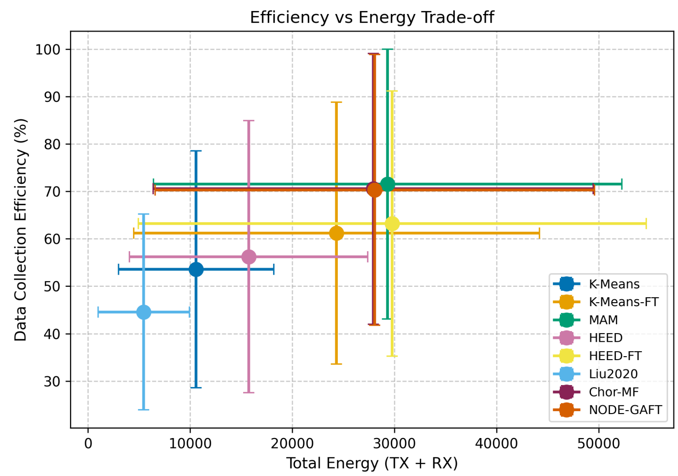


Fig. 9 Efficiency vs. Energy trade-off. Upper-left is better (higher DCE, lower energy).

it collects less data and consumes even less energy proportionally. Among the trajectory-aware algorithms, K-Means achieves a median of 0.67. These higher values reflect that K-Means and Liu 2020 perform less multi-hop forwarding. MAM, Chor-MF, and NODE-GAFT have similar medians (0.32, 0.33, and 0.32 respectively), but Chor-MF has the highest among the three.

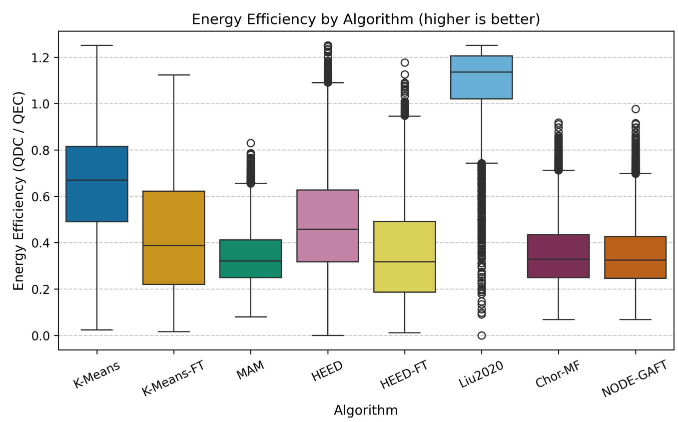


Fig. 10 Energy Efficiency (QDC/QEC) by algorithm. Bigger is better.

Fig. 11 shows how energy efficiency varies with CDR for each algorithm. At low CDR values, all algorithms achieve relatively high energy efficiency as most nodes are within the UAV swarm's direct range and less mesh forwarding is needed. As CDR increases, energy efficiency drops for all approaches because more data must traverse more hops to reach the UAV swarm's radio range. However, the ranking remains consistent: Liu 2020 and K-Means achieve the highest EEff (due to low mesh network usage), while MAM, Chor-MF and NODE-GAFT achieve comparable but lower EEff values due to their more aggressive data forwarding strategies.

Fig. 12 shows the relative DCE+ improvement over K-Means. NODE-GAFT achieves 30.6%, Chor-MF 31.2%, and MAM 36.1%. The results indicate that the improvement over K-Means is significant for all three trajectory-aware approaches, confirming the importance of trajectory awareness in improving data collection efficiency for fixed trajectory scenarios.

Table 6 presents a head-to-head paired comparison of energy efficiency (EEff) between Chor-MF and MAM. Chor-MF out-

 DOI: https://doi.org/10.65218/jius.2026.124

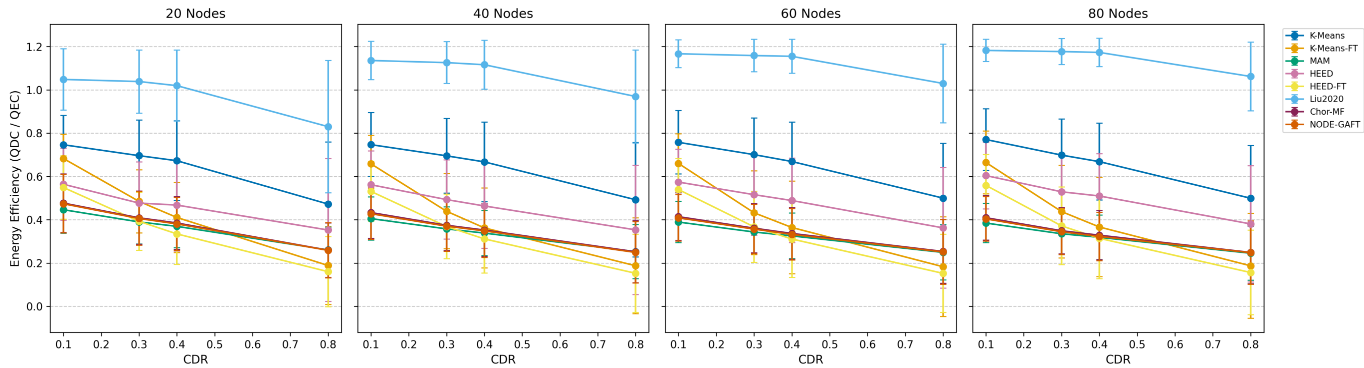


Fig. 11 Energy Efficiency vs. CDR for all algorithms. Bigger is better.

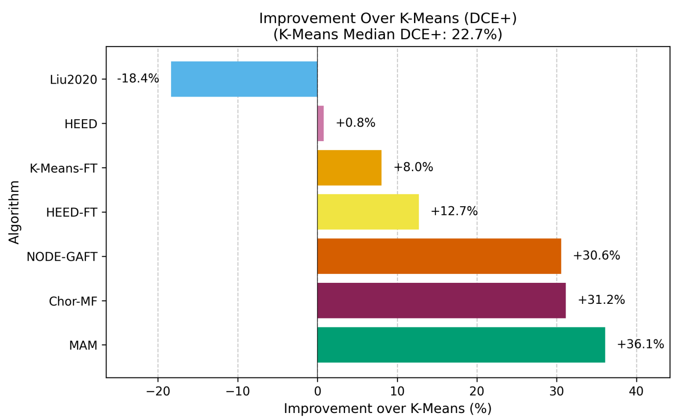


Fig. 12 Relative DCE+ improvement over K-Means (SDCC-normalized). Bigger is better.

performs MAM in energy efficiency in 70.3% of non-tied simulation runs (8,053 wins vs 3,403 losses, Wilcoxon $p < 0.001$, $\hat{A}_{12} = 0.659$, medium effect). This is the clearest metric where our approach consistently outperforms MAM in the results we reported so far. The advantage is strongest at low CDR: at 10% CDR, Choreographed-MaxFlow achieves higher energy efficiency in 84.1% of non-tied runs ($\hat{A}_{12} = 0.792$, large effect). At 80% CDR, the win rate drops to 43.3% as the constrained topology limits optimization opportunities.

Table 6 Head-to-head Energy Efficiency: Chor-MF vs MAM, per CDR.

CDR	Wins	Losses	Ties	Win%	$\hat{A}_{12}$
10%	2,772	524	704	84.1	0.792
30%	2,267	687	1,046	76.7	0.719
40%	1,959	810	1,231	70.7	0.670
80%	1,055	1,382	1,563	43.3	0.456
Overall	8,053	3,403	4,544	70.3	0.659

Among the 16,000 paired runs, 87.5% had DCE values within 5 pp of each other. Chor-MF consumes less total energy than MAM in 72.8% of cases (median saving: 4.9%), confirming that pre-computed routing avoids unnecessary multi-hop transmissions.

Fig. 13 compares energy efficiency among the three top-performing algorithms at different CDR levels. At CDR = 10% and 30%, our approaches have visibly higher median energy efficiency. At CDR = 80%, the distributions converge and MAM's median slightly exceeds ours.

Fig. 14 shows the DCE+ (SDCC-normalized) distribution. The median DCE+ for MAM is 30.85%, while Chor-MF achieves 29.74% and NODE-GAFT 29.60%. The gap between our approaches and MAM is smaller under DCE+ than under raw DCE, indicating that our approaches are more competitive after accounting how much data can actually be collected in each scenario (normalizing by SDCC).

Fig. 15 shows the energy-efficiency trade-off between our approaches and MAM per run, with DCE difference on the x-axis and energy difference on the y-axis (lower-right is better). Most points cluster in the lower-left quadrant (lower DCE and lower energy than MAM), but a significant number fall in the lower-right quadrant, representing scenarios where our approach achieves both higher DCE and lower energy consumption than MAM, which are the scenarios we further investigate in the topology-focused experiments.

D. *Statistical Significance*

We follow the practical guidelines of Arcuri and Briand [7] for assessing randomized algorithms: (1) non-parametric Mann–Whitney U tests for pairwise comparisons; (2) Wilcoxon signed-rank tests on paired same-map and same-speed runs; (3) Vargha–Delaney $\hat{A}_{12}$ effect sizes [31] (negligible: $|\hat{A}_{12} - 0.5| < 0.06$; small: 0.06–0.14; medium: 0.14–0.21; large: ≥ 0.21). With $n = 16{,}000$ runs per algorithm, we plenty of samples to assess statistical significance [7].

The results from those tests indicate that: (1) our approaches (Choreographed-MaxFlow and NODE-GAFT) significantly outperform all trajectory-agnostic approaches with small to large effect sizes; (2) MAM vs our approaches is statistically significant but with negligible effect sizes on DCE ($\hat{A}_{12} = 0.512$) and DCE+ ($\hat{A}_{12} = 0.506$), indicating practically equivalent performance; (3) Choreographed-MaxFlow and NODE-GAFT are statistically indistinguishable across DCE ($p = 0.398$), DCE+ ($p = 0.595$), and energy ($p = 0.350$); and (4) for energy efficiency, our approaches have a statistically significant advantage over MAM in 70.3% of paired runs ($\hat{A}_{12} = 0.659$, medium effect).

Given that MAM and our approaches have similar overall performance on randomized topologies, yet differ in design (reactive routing vs pre-computed route schedules), we hypothesized that the random map generator does not produce the specific topology patterns that favor pre-computed routing. The C-shape

 DOI: https://doi.org/10.65218/jius.2026.124

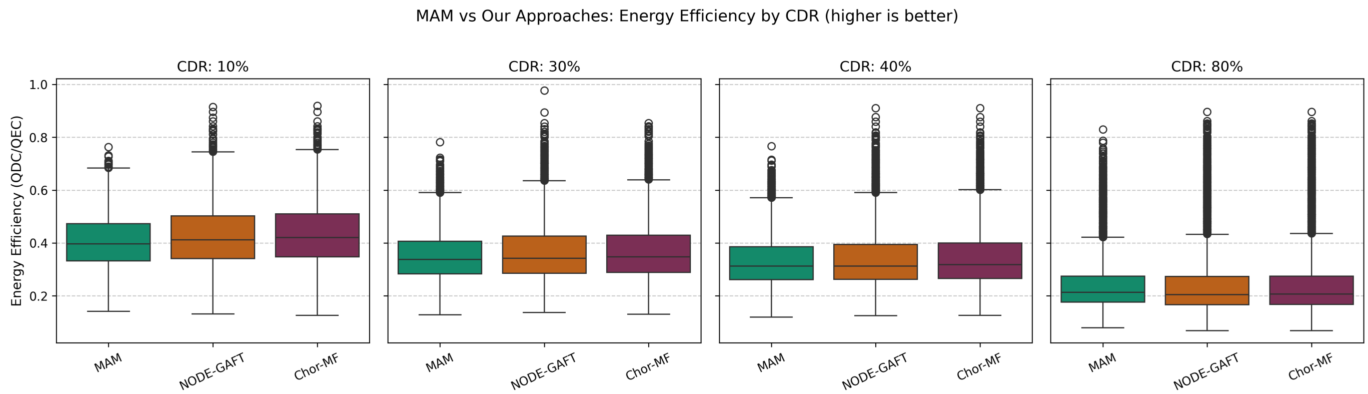


Fig. 13 Energy Efficiency by CDR: MAM, Chor-MF, and NODE-GAFT. Bigger is better.

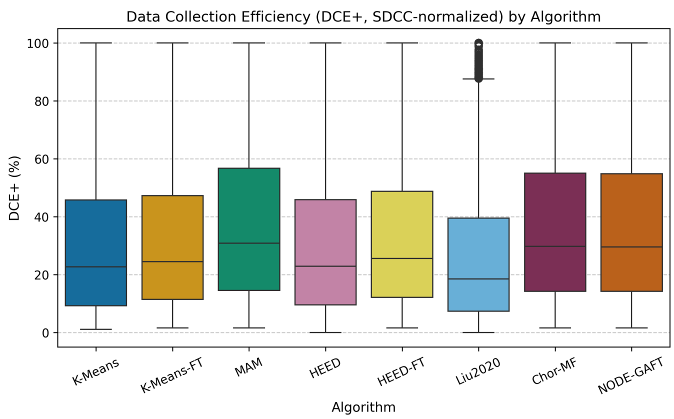


Fig. 14 DCE+ (SDCC-normalized) distribution by algorithm. Bigger is better.

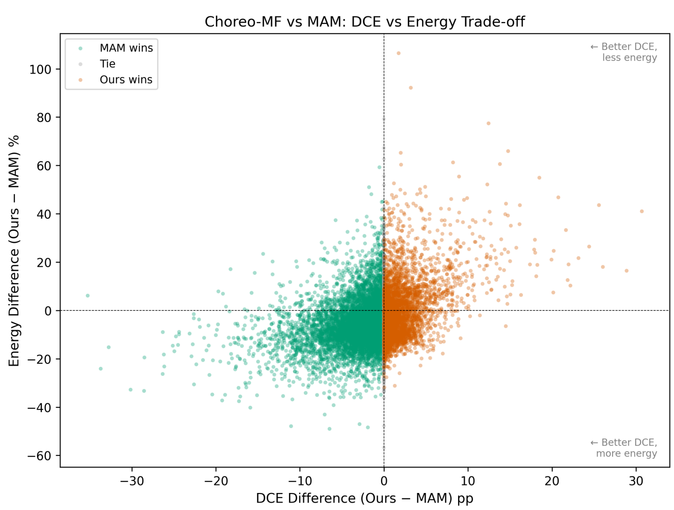


Fig. 15 DCE vs. Energy trade-off: Chor-MF individual runs compared to MAM. Lower-right is best.

patterns observed in winning runs (Section 5.2) motivated the topology-focused experiments we present next.

VI. *Topology-Focused Experiments*

The randomized simulations we presented in the previous sections generate maps by randomly distributing nodes across the map area, with a few restrictions to make the network a valid mesh network and respect the coverage deficit ratio (CDR) parameter. While this approach covers a vast amount of scenarios, we believe it makes it unlikely to generate specific topological patterns that may favor one algorithm over another, and which would also be interesting to understand and discuss.

As discussed in Section 5.2, manual inspection of the runs where Choreographed-MaxFlow and NODE-GAFT outperform MAM revealed a recurring pattern: the ground network forms an elongated or curved structure where the UAV swarm's coverage band intersects different parts of the network at different times when UAVs fly over it. We designed custom topologies to exaggerate this pattern and investigate it by running similar simulation and metrics analysis.

Our hypothesis is that the main factor that this pattern causes is the presence of what we call ground-air temporal hiatuses between contact regions. We define those as periods after the UAV swarm first encounters the ground network (AMEnc MEET phase) in which no ground nodes are within radio range of the UAV swarm for some significant amount of time, but eventually the swarm collects data from more nodes of the same ground network. In such scenarios, data stored on out-of-range nodes must be routed through the mesh network in advance to reach a contact region before the UAVs pass it. A reactive algorithm like MAM can only begin pulling data once the UAVs are already within range of a contact region, whereas Choreographed-MaxFlow and NODE-GAFT, with their pre-computed flow scheduling choreography, can begin routing data toward a future contact point while the UAVs are still traversing an empty region after the first contact window. We believe that the bigger the temporal separation between contact regions, the greater the advantage of the pre-computed routing approach.

A. *Custom Topology Design*

Fig. 16 illustrates both custom topologies. Blue nodes are within the UAV swarm's radio coverage band (orange shaded area); red nodes are outside. The green arrow indicates the UAV trajectory direction.

Both the "inverted-C shape" (Fig. 16(b)) and the "3-shape" (Fig. 16(a)) topologies we designed (handcrafted) contain 80 sensor nodes with a radio range of 100 map units, the same UAV swarm formation (5 UAVs flying in "V-shape"), and form a fully connected mesh network. We more formally define the topologies and scenarios we aim to cover with this design by establishing the following characteristics: (i) a significant portion

 DOI: https://doi.org/10.65218/jius.2026.124

of nodes are outside the UAV radio coverage band (high CDR), (ii) the nodes within radio range are distributed across multiple, temporally separated contact regions along the UAV trajectory, separated by a significant temporal hiatus due to the UAV swarm trajectory and speed and (iii) the temporal hiatuses between contact regions force data to traverse the mesh network before the UAVs arrive at the next contact point in order to better utilize the available bandwidth. This forces data routing decisions to be made ahead of time if we want to collect more data, which is precisely the scenario where we expect pre-computed routing / choreographed approaches (Choreographed-MaxFlow or NODE-GAFT) to outperform reactive routing (MAM) in terms of DCE and EEff.

We generate both topologies by placing nodes along a skeleton of connected line segments defining the desired shape, with configurable spacing and random jitter. We add padding nodes per region until the target count ($N = 80$) is reached, with configurable density bias. The algorithm verifies full mesh connectivity and retries with adjusted jitter if verification fails. Algorithm 3 describes the generation process.

Algorithm 3 Custom Topology Generation

Require: Skeleton segments $\mathcal{P} = \{(p_1, p_2, \delta_i, j_i)\}$ (endpoints, spacing, jitter); target node count N; radio range ϕ; density bias $\mathcal{B}$

```
S ← ∅
for each segment (p1, p2, δ, j) ∈ P do
    L ← ||p2 − p1||, k ← ⌊L/δ⌋
    for i = 0 to k − 1 do
        (x, y) ← (1 − i/k) · p1 + (i/k) · p2 + Uniform(−j, j)
        S ← S ∪ {(x, y)}
    end for
end for
while |S| < N do
    Pick region r from B with biased probability
    (x, y) ← random point within region r
    S ← S ∪ {(x, y)}
end while
G ← graph where (i, j) ∈ E iff dist(S_i, S_j) ≤ φ
if G is not connected then
    Retry with adjusted jitter
end if
return S, G
```

B. 3-Shape Topology

The 3-shape topology arranges nodes in the shape of the number "3," with three horizontal lines extending into the UAV coverage band at the top ($y \approx 80$), middle ($y \approx 300$), and bottom ($y \approx 520$) of the map. A vertical line at $x \approx 550$ (outside the coverage band) connects the horizontal lines. The UAV swarm first encounters the top horizontal line, then the middle, and then the bottom. This creates three temporally separated contact windows and two hiatuses between them. When the UAV swarm is in a hiatus period (flying over the region between two horizontal lines), no ground nodes are within radio range, and data must be pre-routed through the mesh toward the next contact region to be collected.

C. Inverted-C Topology

The Inverted-C topology arranges nodes in an inverted "C" shape, with its tips reaching the coverage band at the top and bottom of the map and most of its arc extending to the right ($x \approx 500$–640), well outside of the UAV swarm's radio coverage. The node distribution is intentionally asymmetric, with the top tip (encountered first by the UAV swarm, first contact region) having low node density, and the bottom tip (last contact region) high node density. This design stresses whether a predictive data collection algorithm can forecast the upcoming contact with a dense region and begin pre-routing data toward the bottom tip while the UAVs are at communication hiatus. This design creates one hiatus and two contact windows. The asymmetric density distribution further penalizes reactive routing approaches like MAM, which cannot begin routing data toward the dense bottom tip until the UAVs arrive there.

D. Custom Topology Results

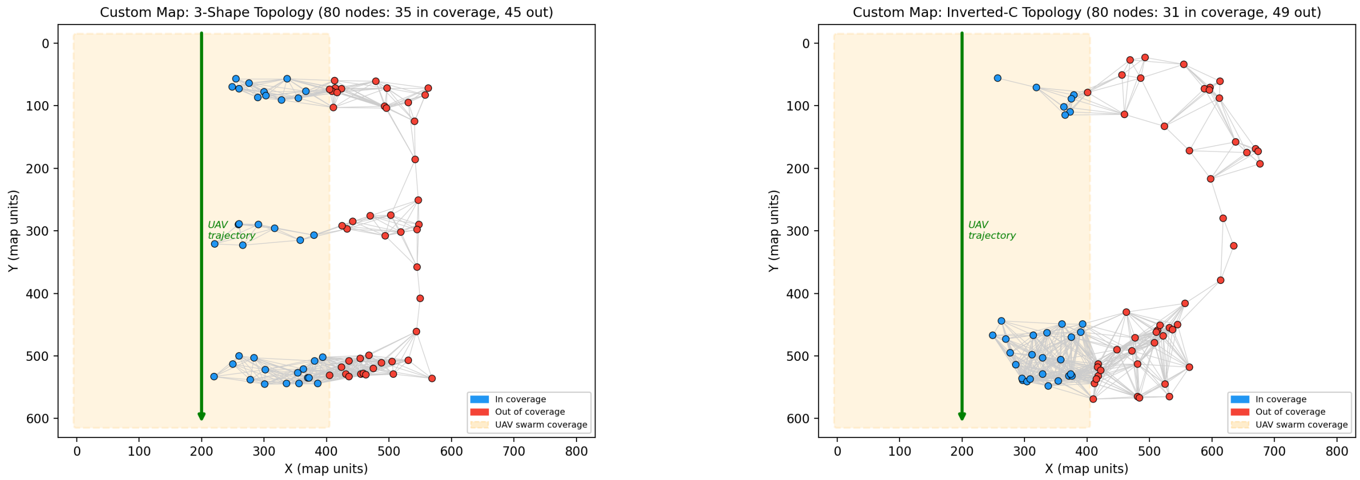
We ran all eight algorithms on both custom topologies with five UAV speeds (0.02, 0.05, 0.1, 0.2, and 0.4) and three initial data distributions (100, 250, and 500), using the same units and system model as the main experiments we conducted and presented in previous sections. This resulted in 240 simulation runs. Table 7 presents the results for the top three algorithms (Choreographed-MaxFlow, NODE-GAFT, and MAM) in those 240 simulations.

For the 3-Shape topology, our approaches (Choreographed-MaxFlow and NODE-GAFT) outperform MAM in 10 out of 15 configurations of different speed and initial data distribution values. The most significant win is at speed 0.02 with data=500, where Choreographed-MaxFlow reaches 87.2% DCE versus MAM's 73.2% (+14.0 pp gain). NODE-GAFT also shows promising results: at speed 0.10 with data=100 it reaches 63.2% versus MAM's 51.7% (+11.5 pp). MAM performs better than our approaches only in two scenarios, one with the fastest speed (0.40 with data=250) and the other with the lowest speed (0.02 with data=250). Our approach surpassed MAM even in a lowest speed scenario: speed 0.20 with data=100, where Choreographed-MaxFlow (39.2% DCE) outperforms MAM (36.0% DCE) by +3.2 pp.

For the Inverted-C topology scenarios, the asymmetric node density produced even bigger wins for our approaches. The largest gain is at speed 0.02 with data=500, where Choreographed-MaxFlow reaches 78.2% DCE versus MAM's 63.6% DCE, which is a +14.6 pp advantage. At speed 0.02 with data=250, NODE-GAFT achieves 90.6% versus MAM's 77.5% (+13.1 pp). At speed 0.05, our approaches win for every initial data parameter: with data=100, NODE-GAFT (82.8%) outperforms MAM (80.3%); with data=250, Choreographed-MaxFlow (62.3%) outperforms MAM (54.9%) by +7.4 pp; with data=500, Choreographed-MaxFlow (53.4%) outperforms MAM (46.5%) by +6.9 pp. Since the majority of data is stored near the bottom tip of the shape, it forces MAM's reactive routing to pull data downward only after the UAVs arrive at the bottom contact region, whereas our pre-computed approaches schedule that behavior in advance. At higher speeds (0.20–0.40), the advantage narrows since contact windows are shorter (e.g., at speed 0.40 with data=100, MAM wins with 23.8% vs Chor-MF's 23.6%, a negligible +0.2 pp gain).

Regarding energy efficiency, for the C-Shape at speed 0.02 with data=100, all three algorithms achieve 100% DCE, but our approaches use 18–21% less energy than MAM (66,025 and 75,580 vs 80,587 energy units). For the 3-Shape at speed 0.02 with data=500, Choreographed-MaxFlow collects 14 pp more

 DOI: https://doi.org/10.65218/jius.2026.124



(a) 3-Shape topology.

(b) Inverted-C topology.

Fig. 16 Custom topologies with temporally separated contact regions. Blue: in-range nodes; Red: out-of-range nodes; Orange: UAV coverage band; Green arrow: UAV trajectory.

Table 7 Custom topology results: DCE (%) and Total Energy for Chor-MF, NODE-GAFT, and MAM per UAV speed and initial data level. Bold: best DCE per row. Green : our approaches outperform MAM.

			DCE (%)			Total Energy		
Map	Speed	Data	Chor-MF	N-GAFT	MAM	Chor-MF	N-GAFT	MAM
3-Shape topology								
3-Shape	0.02	100	100.0	100.0	100.0	65,581	69,064	62,965
3-Shape	0.02	250	87.9	89.3	**92.9**	91,937	119,025	89,921
3-Shape	0.02	500	**87.2**	79.2	73.2	97,762	98,292	75,678
3-Shape	0.05	100	66.8	**76.4**	72.6	50,866	58,014	49,552
3-Shape	0.05	250	**65.3**	60.6	58.0	51,035	65,313	51,785
3-Shape	0.05	500	**59.9**	54.1	53.5	49,824	44,165	41,328
3-Shape	0.10	100	54.4	**63.2**	51.7	38,554	41,106	36,766
3-Shape	0.10	250	**49.2**	49.0	42.4	42,827	47,413	36,805
3-Shape	0.10	500	**45.4**	45.2	44.5	33,667	37,519	28,245
3-Shape	0.20	100	**39.2**	38.9	36.0	27,259	26,918	25,946
3-Shape	0.20	250	**34.0**	31.7	33.6	30,160	29,756	28,213
3-Shape	0.20	500	**33.2**	32.9	32.6	20,883	18,993	18,596
3-Shape	0.40	100	27.8	**29.0**	27.4	18,643	18,424	18,710
3-Shape	0.40	250	23.6	23.3	**24.2**	22,125	21,785	22,239
3-Shape	0.40	500	19.3	19.3	19.3	11,210	10,946	10,853
Inverted-C topology (asymmetric: sparse top tip, dense bottom tip)								
C-Shape	0.02	100	100.0	100.0	100.0	66,025	75,580	80,587
C-Shape	0.02	250	86.8	**90.6**	77.5	105,115	106,347	103,793
C-Shape	0.02	500	**78.2**	75.9	63.6	118,270	108,265	95,735
C-Shape	0.05	100	74.0	**82.8**	80.3	45,609	56,947	54,998
C-Shape	0.05	250	**62.3**	60.7	54.9	62,679	68,530	52,573
C-Shape	0.05	500	**53.4**	49.3	46.5	58,755	52,325	40,706
C-Shape	0.10	100	47.0	50.1	**57.1**	34,995	39,518	38,431
C-Shape	0.10	250	**47.9**	42.1	41.8	43,364	45,907	36,802
C-Shape	0.10	500	**37.2**	35.8	37.0	33,163	31,295	25,568
C-Shape	0.20	100	37.2	32.9	**37.9**	24,376	25,699	26,191
C-Shape	0.20	250	**31.5**	29.4	29.9	30,494	31,482	27,826
C-Shape	0.20	500	25.6	25.5	25.6	19,072	18,262	15,563
C-Shape	0.40	100	23.6	23.7	**23.8**	16,857	17,177	17,378
C-Shape	0.40	250	**21.1**	21.1	20.6	19,390	19,978	20,374
C-Shape	0.40	500	13.8	13.8	13.8	8,302	7,945	8,323

 DOI: https://doi.org/10.65218/jius.2026.124

data than MAM while consuming 29% more energy, indicating that the additional data collection outweighs the energy cost.

These limited experiment results indicate that UAV swarms performing opportunistic data collection from WMNs with certain topology patterns may collect more data and with a smaller energy footprint if they use our proposed Choreographed MaxFlow and NODE-GAFT approaches instead of MAM. The topology patterns we tested were two elongated networks with temporally separated contact regions. On the 3-Shape topology, our approaches win in 10 of 15 configurations, with Choreographed-MaxFlow reaching +14.0 pp over MAM at speed 0.02 with data=500. On the asymmetric C-Shape topology, our approaches win in 9 of 15 configurations, with gains up to +14.6 pp (speed 0.02, data=500).

With the random map generator from the original 128,000 randomized experiments, it seems unlikely that such topology patterns appear, but we can find those in real-world scenarios such as sensor networks deployed along a curved bayside (depending on their size, curvature and the UAVs trajectory, they are an example of a natural formation in form of a "C"), or pipeline corridors. The results above, however, are based on a single map per topology. To validate that the advantages we observed are not limited to those specific node positions with specific jitter values and node distributions, we conduct a larger randomized experiment that follows those topological patterns.

E. Randomized Topology Validation

We generated maps for each topology shape (3-Shape and C-Shape) using the procedure we described in the previous section, and used two network sizes (60 and 80 nodes) and a repetition parameter of 16, yielding 64 distinct maps. Each variant randomizes the exact node positions by adding jitter, producing different ground network mesh connectivity graphs each time. We then ran all eight algorithms on each map at three UAV speeds (0.05, 0.1, and 0.2) and three initial data distributions (100, 250, and 500 data units per node), resulting in 4,608 simulation runs. This allows us to assess whether the pre-computed routing advantage we observed in the handcrafted runs generalizes across random topologies with similar patterns and different network sizes, or whether it is specific to some particular condition unique to our handcrafted runs.

Fig. 17 shows all eight algorithms' mean DCE on the randomized custom topologies, confirming a similar ranking observed in the 128,000 randomized experiments: MAM, Chor-MF, and NODE-GAFT are the top performers, followed by the FT adaptations, then the trajectory-agnostic algorithms. However, in this ranking, Chor-MF slightly outperforms MAM, in both topologies.

Fig. 18 shows DCE boxplots across all randomized custom topology runs. On the 3-Shape, Chor-MF and NODE-GAFT have higher medians and tighter interquartile ranges than MAM. On the C-Shape, the three approaches have similar medians, indicating that the asymmetric node distribution may have a negative effect in some of the parameter combinations we evaluated (e.g., lower or higher speeds and data distribution, that would make MAM perform more competitively in some runs).

Fig. 19 shows DCE violin plots for Chor-MF, NODE-GAFT, and MAM. Chor-MF has a slightly higher median for both topologies, with a wider tail toward higher DCE values. Fig. 20 confirms the same pattern under DCE+ normalization.

Fig. 21 shows mean DCE vs. UAV speed: DCE decreases from roughly 60–70% at speed 0.05 to 25–35% at speed 0.2. Chor-MF's advantage over MAM is consistent across speeds on the 3-Shape but narrower on the C-Shape.

Fig. 22 presents Chor-MF's win rate against MAM by topology and scenario as a heatmap. For the 3-Shape topology, Chor-MF's win rate is consistently above 60%, reaching 91% at speed 0.20 with data=500.

Fig. 23 presents NODE-GAFT's win rate against MAM. NODE-GAFT shows the highest win rate at speed 0.20 with data=500 (88%) on the 3-Shape topology, but its performance is less consistent across scenarios than Chor-MF, with win rates varying from 44% to 88%.

Fig. 24 presents a heatmap of median energy efficiency (EEff) by algorithm and scenario (speed and initial data distribution) for both topologies. MAM achieves higher energy efficiency than our approaches in most scenarios. The gap is larger at higher speeds and lower initial data distribution levels, where brief contact windows limit how much additional data the pre-computed schedule can deliver. Energy efficiency is higher at higher initial data levels (data distribution), since the shorter contact windows at higher speeds lead to less data being forwarded between ground nodes.

Following the same methodology as the main experiments, we ran statistical significance tests using Mann-Whitney U tests and Wilcoxon signed-rank paired tests for the randomized custom topology results. Table 8 presents those results.

Table 8 Paired statistical tests on randomized custom topologies (4,608 runs). N=negligible, S=small, M=medium, L=large effect.

Topology	Comp.	W	L	W%	$\hat{A}_{12}$	p	Eff.
DCE							
3-Shape	C-MF	198	89	69.0	0.690	<.001	M
3-Shape	N-G	171	117	59.4	0.594	.001	S
C-Shape	C-MF	154	132	53.8	0.538	.008	N
C-Shape	N-G	133	153	46.5	0.465	.309	N
Energy Efficiency (EEff)							
3-Shape	C-MF	46	140	24.7	0.330	<.001	M
3-Shape	N-G	51	145	26.0	0.333	<.001	M
C-Shape	C-MF	56	144	28.0	0.347	<.001	M
C-Shape	N-G	52	164	24.1	0.253	<.001	L

For the randomized 3-Shape topologies, Choreographed-MaxFlow outperforms MAM in 198 out of 287 non-tied paired runs (69.0% win rate, Wilcoxon $p < 0.001$, $\hat{A}_{12} = 0.690$, medium effect) for DCE, with a mean overall DCE advantage of $+1.89$ pp. NODE-GAFT also significantly outperforms MAM on this topology, with a 59.4% win rate (Wilcoxon $p = 0.001$, $\hat{A}_{12} = 0.594$, small effect, $+0.83$ pp). This indicates that the three temporally separated contact windows of the 3-shape topologies consistently favor choreographed routing (our approaches).

For the randomized C-Shape topologies, Choreographed-MaxFlow outperforms MAM in 53.8% of non-tied paired runs (Wilcoxon $p = 0.008$, $\hat{A}_{12} = 0.538$, negligible effect), with a $+0.77$ pp increase. NODE-GAFT does not significantly outperform MAM on this topology (46.5% win rate, Wilcoxon $p = 0.309$, $\hat{A}_{12} = 0.465$, negligible effect), suggesting that the C-shape topology with two contact regions offers a less con-

DOI: https://doi.org/10.65218/jius.2026.124

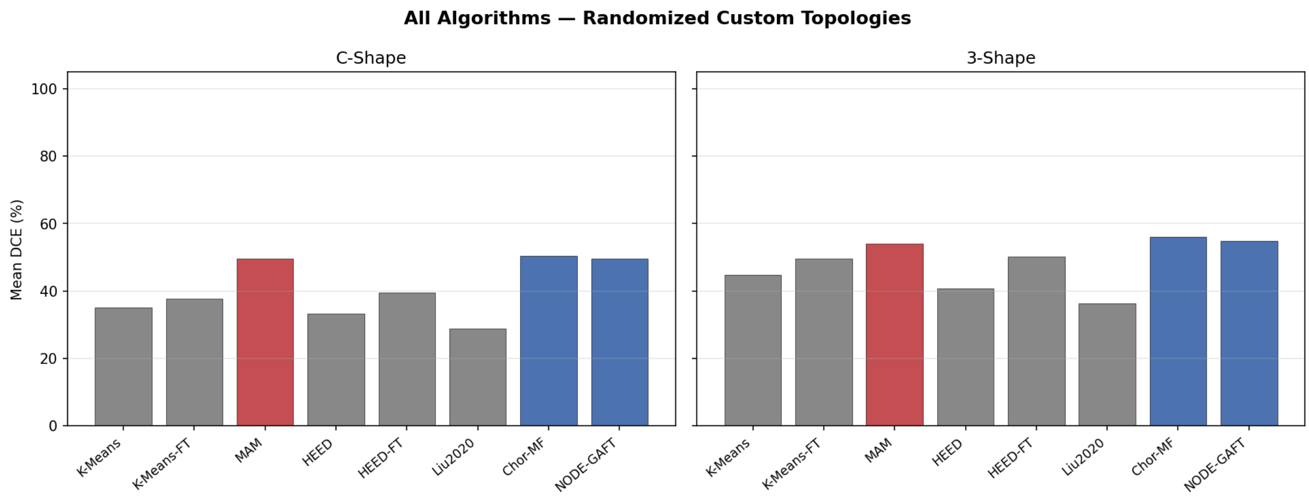


Fig. 17 Mean DCE for all eight algorithms on randomized custom topologies.

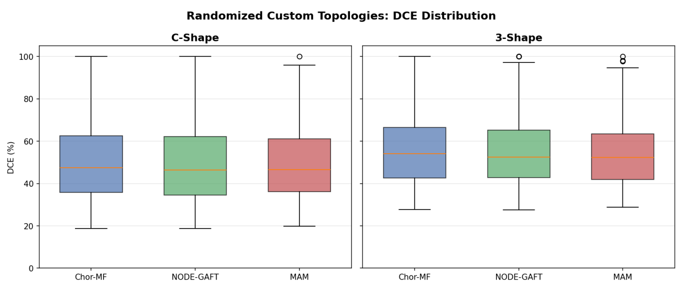


Fig. 18 DCE boxplots per algorithm on randomized custom topologies with subplots per topology.

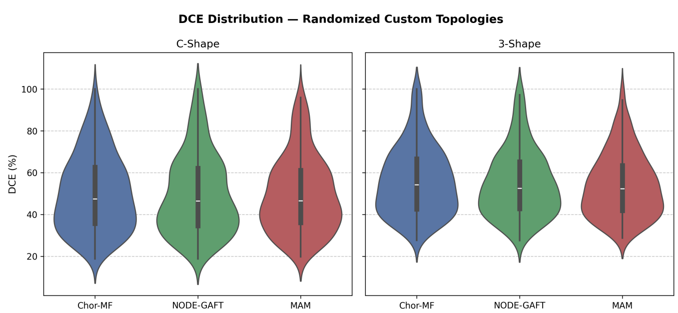


Fig. 19 DCE violin plots per approach on randomized custom topologies.

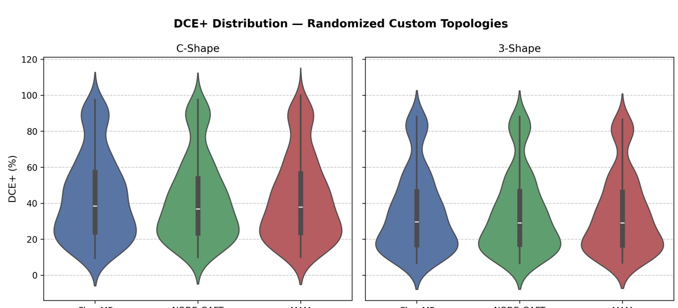


Fig. 20 DCE+ violin plots per approach on randomized custom topologies.

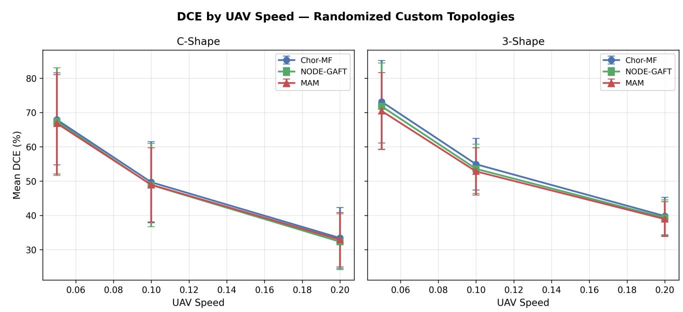


Fig. 21 Mean DCE by UAV speed on randomized custom topologies. Error bars show standard deviation.

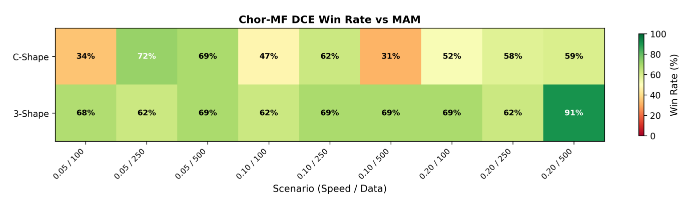


Fig. 22 Chor-MF DCE win rate vs MAM by topology and scenario (speed / data), color-coded from red to green.

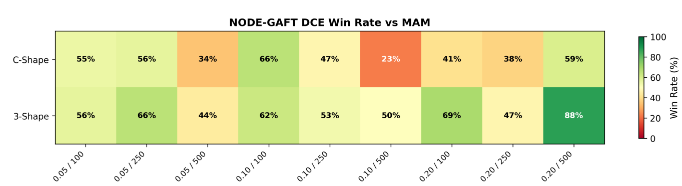


Fig. 23 NODE-GAFT DCE win rate vs MAM by topology and scenario, color-coded from red to green.

sistent advantage than the 3-shape with three contact regions.

The randomized topology experiments support the hypothesis that we formulated after the single handcrafted map simulations, that topologies with temporally separated contact regions (where there are different contact regions separated by out-of-range hiatuses) favor pre-computed choreographed routing (our approaches) over reactive routing (MAM) more consistently than in the fully random 128,000 map simulations from previous sections.

DOI: https://doi.org/10.65218/jius.2026.124

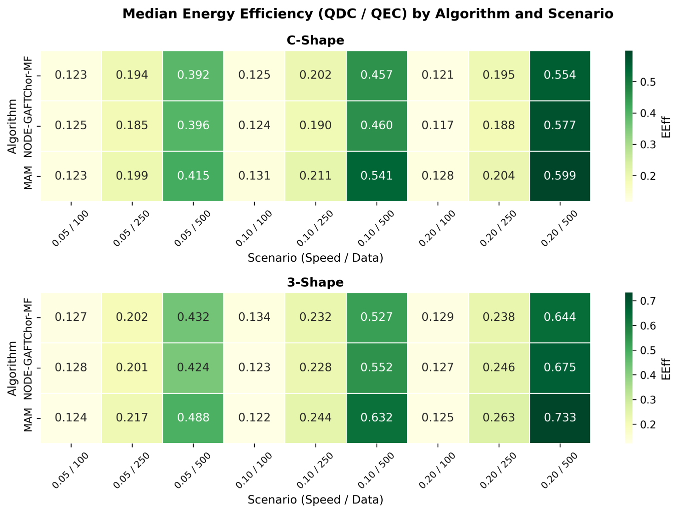


Fig. 24 Median Energy Efficiency (EEff) heatmap by algorithm and scenario on randomized custom topologies.

F. Temporal Hiatus Detection

Our experiments indicate that our pre-computed routing approaches (Choreographed-MaxFlow and NODE-GAFT) outperform reactive routing (MAM) in several cases when the ground network and the UAV trajectory leads to temporal hiatuses (gaps along the UAV trajectory where no sensor nodes are within radio coverage) after the UAV swarm first encounters the ground network. This raises an important question for a real-world test: since using Chor-MF and NODE-GAFT yield better results in many cases (up to 91% of cases for speed=0.2 data=500 across both topologies, and close to 70% overall for the 3-Shape topology) where there are those temporal hiatuses, given an arbitrary network topology and a known UAV trajectory, can we automatically detect whether temporal hiatuses exist to decide which routing approach should be applied on-the-fly?

We propose a simple detection algorithm (Algorithm 4) that aims to answer this question by analyzing the ground mesh nodes position relative to the UAV swarm's coverage band.

The detection algorithm works in three steps: (1) partition mesh nodes by coverage (detect which nodes are within the UAV swarm's radio range along its trajectory, and which ones are not); (2) sort by trajectory direction; (3) scan for consecutive gaps exceeding a threshold of the radio range $\tau = 1.5 \times \phi$ (where ϕ is the radio range). We then flag the topology as Has-Hiatus if at least one gap is detected, or return No-Hiatus otherwise. We chose the threshold factor of 1.5 empirically; it must be large enough to distinguish true hiatuses from normal spacing between nodes, yet small enough to detect the gaps present in topologies like the 3-Shape (in our case, with gaps of 170–350 map units with $\phi = 100$).

We designed the detection algorithm to be executed at the moment the UAV swarm first establishes contact with the ground network and receives node position information. If the detector returns Has-Hiatus, the swarm can instruct the ground network to switch from a reactive routing strategy (e.g., MAM) to a pre-computed approach (Choreographed-MaxFlow or NODE-GAFT), which our experiments show yields higher DCE in such topologies in many cases. This makes the algorithm selection topology-adaptive, in which MAM is used as the default routing approach (since it performs well in most random deployments), but when the UAV swarm detects a topology pattern which may favor pre-computed routing, it reconfigures the ground network routing strategy.

Algorithm 4 Temporal Hiatus Detection

Require: Sensor nodes S, UAV coverage band $[x_{\min}, x_{\max}]$, radio range ϕ

$\tau \leftarrow 1.5 \times \phi$ {hiatus threshold}
$S_{\text{in}} \leftarrow \{n \in S \mid x_{\min} \leq n.x \leq x_{\max}\}$
$S_{\text{out}} \leftarrow S \setminus S_{\text{in}}$
Sort S_{in} by y-coordinate (ascending, along trajectory)
hiatuses $\leftarrow 0$
for $i = 1$ to $|S_{\text{in}}| - 1$ **do**
  if $S_{\text{in}}[i].y - S_{\text{in}}[i-1].y > \tau$ **then**
    hiatuses $\leftarrow$ hiatuses $+ 1$
  end if
end for
if hiatuses ≥ 1 **then**
  return HAS-HIATUS
else
  return NO-HIATUS
end if

We implemented the detector and validated it against 128 maps: the 64 randomized custom topology maps from Section 6.5 (expected Has-Hiatus) and 64 randomly selected maps from the main 128,000-run experiments (expected No-Hiatus, as uniformly distributed nodes rarely produce large coverage gaps). Table 9 has the results of those validations.

Table 9 Temporal Hiatus Detector validation (128 maps). Threshold: $\tau = 1.5 \times \phi$ where $\phi = 100$ map units.

Metric	Value	Metric	Value
True Positives (TP)	62	Accuracy	98.4%
True Negatives (TN)	64	Precision	100.0%
False Positives (FP)	0	Recall	96.9%
False Negatives (FN)	2	F1 Score	98.4%

The detector achieves 98.4% accuracy with perfect (100%) precision (no false positives) and 96.9% recall. There were only two false negatives, which were C-Shape topologies with 60 nodes where the C arc nodes fell close to the coverage boundary. This actually seems like correct behavior, since we analyzed the map instances and verified they lacked a strong temporal hiatus.

The custom topology experiments and the temporal hiatus detector point toward a promising practical direction: a hybrid routing strategy that uses the detector at the moment of ground-air encounter (AMEnc MEET phase) to decide whether to apply Choreographed-MaxFlow (when temporal hiatuses are detected) or MAM (in the general case), combining the strength of both approaches to achieve higher DCE across a wider range of scenarios.

VII. Discussion

Our results reveal several important insights about data collection algorithm design for fixed-trajectory UAV scenarios.

The single most important design decision that our results indicate is whether the algorithm considers the UAV trajectory. All trajectory-aware approaches (MAM, Choreographed-MaxFlow, NODE-GAFT, K-Means-FT, HEED-FT) significantly outperform trajectory-agnostic ones (K-Means, HEED, Liu 2020).

DOI: https://doi.org/10.65218/jius.2026.124

Traditional clustering approaches (K-Means, HEED) suffer from the limitation that cluster heads may not be optimally positioned along the UAV trajectory, leading to low data collection efficiency (DCE). K-Means selects cluster heads purely based on spatial clustering, while HEED considers residual energy and communication cost, but neither accounts for which nodes the UAVs will actually fly over. Our fixed-trajectory adaptations (K-Means-FT, HEED-FT) address this by constraining cluster head selection to nodes within the UAV swarm's radio range along the trajectory, ensuring that the selected cluster heads are visited by the UAVs. Even this simple adaptation produces significant improvements (K-Means to K-Means-FT: +8 pp; HEED to HEED-FT: +7 pp), validating the hypothesis that trajectory awareness is beneficial for all approaches under the fixed-trajectory constraint.

The unpaired Mann-Whitney U tests show that MAM vs Choreographed-MaxFlow is statistically significant for DCE ($p < 0.001$) but with a negligible effect size ($\hat{A}_{12} = 0.512$)—both share a high median DCE. Under DCE+ (SDCC-normalized), the difference is statistically significant ($p = 0.045$) but with a negligible effect size ($\hat{A}_{12} = 0.506$), confirming that the two approaches are practically equivalent in data collection performance when accounting for scenario differences.

Our evaluation on each 16,000 same-map runs reveal more detail: MAM wins in 7,296 runs vs Choreographed-MaxFlow's 3,939 wins, with 4,765 ties (29.8%). The mean DCE difference is only -1.01 percentage points. MAM's advantage is present across all CDR levels, with Choreographed-MaxFlow's win rate among non-tied pairs ranging from 42.2% (CDR = 10%) to 30.4% (CDR = 80%). This pattern suggests that MAM's reactive routing consistently has a slight advantage in randomized topologies, but the gap is small as shown by the negligible effect sizes.

In contrast, the topology-focused experiments show a different pattern. For the 3-Shape topology with three temporally separated contact regions, Choreographed-MaxFlow wins 69% of paired runs (Wilcoxon $p < 0.001$, $\hat{A}_{12} = 0.690$, medium effect). This significant difference demonstrates that the optimal algorithm choice depends fundamentally on the topology structure.

The C-Shape topology with asymmetric density illustrates an important limitation of reactive routing: MAM cannot begin routing data toward the dense bottom tip until the UAVs arrive there, leading to unused bandwidth in some scenarios (leading to lower DCE compared to Choreographed-MaxFlow). Choreographed-MaxFlow avoids this by scheduling data movement towards the next contat region during the hiatus period, effectively "pre-positioning" data for the next contact window.

The 3-Shape topology (three contact regions, two hiatuses) shows a stronger and more consistent advantage for Choreographed-MaxFlow (69% win rate, medium effect size) compared to the C-Shape (two contact regions, one hiatus; 53.8% win rate, negligible effect size). This suggests that pre-computed routing benefits compound with additional temporal separations, as each hiatus provides an opportunity for data pre-positioning that reactive routing cannot leverage.

As observed in the handcrafted topology experiments (Table 7), at very high UAV speeds (0.40), contact windows become so short that the routing strategy has a much smaller impact (all algorithms converge to similar DCE values). The advantage of pre-computed routing is more evident at slow to medium speeds (0.02–0.10), where longer contact windows allow the choreographed schedule to deliver substantially more data, although higher initial data volumes also tend to favor pre-computed routing at higher speeds (e.g., 0.20), as the routing problem becomes more constrained and optimization has greater impact.

Our research identified several important trade-offs for UAV-WSN data collection system design. Higher data collection efficiency requires more radio transmissions. Among the high-performing algorithms, MAM consumes median 22,801 energy units vs Choreographed-MaxFlow's 21,950 (3.7% less). However, algorithms that collect less data (Liu 2020, K-Means) also use less energy, creating a false impression of "efficiency" (as it is not a true efficiency gain but rather a consequence of lower data collection performance).

The pre-computed max-flow routing does not merely save energy by collecting less data. Instead, it routes data more efficiently, avoiding unnecessary multi-hop transmissions that MAM's reactive routing performs in certain scenarios. The energy efficiency advantage is more evident in networks with low CDR, where there are multiple possible paths and the flow computation can select the most efficient ones. In networks with high CDR (e.g., 80%), the constrained topology leaves fewer routing alternatives, reducing the number of opportunities for energy efficiency optimization in the scenarios we evaluated.

We also observe that when Choreographed-MaxFlow outperforms MAM on DCE, it tends to consume more energy in those runs (70.8% of DCE-winning runs have higher energy than MAM, mean $+979.8$ energy units which is a very small value proportionally to the overall energy consumption we presented in Table 7). This is not contradictory: to outperform MAM's reactive routing, the mesh network must perform more forwarding (potentially from more distant nodes), which costs energy. The energy efficiency advantage is concentrated in the much larger set of runs where both algorithms collect comparable data—there, Choreographed-MaxFlow achieves the same result, consistently across thousands of randomized maps, with less wasted routing and energy consumption.

For the custom topology experiments, the results are different. For the C-Shape topology at speed 0.02 with data=100, all three algorithms achieve 100% DCE, but our approaches use 18–21% less energy than MAM (66,025 and 75,580 vs 80,587 energy units), a significant difference that favors our approaches. This demonstrates that when the topology favors pre-computed routing, our approaches can achieve both higher DCE and significantly lower energy consumption.

In our simulations, NODE-GAFT produces results that are statistically indistinguishable from Choreographed-MaxFlow, since all nodes start with full energy and results indicate that the energy scaling has negligible effect in a single run. NODE-GAFT is expected to differentiate itself in multi-round scenarios where energy depletion continuously happen and creates heterogeneous energy states across the network, but this evaluation is out of scope and we leave this as future work.

Regarding algorithm complexity, K-Means-FT (61.22% mean DCE) is the simplest trajectory-aware approach, which merely constrains cluster heads to nodes along the UAV trajectory, yet it achieves $\sim$9 pp less than Choreographed-MaxFlow (70.54%), which requires computing maximum flow at each trajectory stage. This gap illustrates the trade-off between implementa-

DOI: https://doi.org/10.65218/jius.2026.124

tion simplicity and data collection performance: for applications where simplicity is paramount (e.g., there are lower CPU and memory resources), even basic trajectory awareness provides significant gains; for applications requiring maximum throughput, the computational cost of Choreographed-MaxFlow may be justified.

While our experiments use a simplified radio model (binary in-range/out-of-range within a fixed radius, without interference), the topological patterns we identified (elongated or curved networks with separated contact regions) are plausible in real-world scenarios. Sensor networks deployed along coastlines, pipeline corridors, river banks, or mountain ridgelines may naturally form elongated and curved structures. When UAV swarms patrol such areas along a fixed route, temporal hiatuses may arise depending on their trajectory, where portions of the network are out of radio range between contact windows.

The temporal hiatus detector provides a lightweight mechanism (requiring only node positions and the UAV trajectory) to determine at runtime (AMEnc MEET phase) whether pre-computed routing is beneficial. This makes it feasible to implement a hybrid strategy where MAM serves as the default routing approach (since our results indicate it performs well on most uniformly distributed deployments), with the system switching to Choreographed-MaxFlow when the detector identifies temporal hiatuses. This hybrid approach would combine the strengths of both approaches in a topology-adaptive manner, which should lead to higher DCE across a wider range of real-world scenarios.

We acknowledge that our simulations have several simplifications: no radio interference or packet loss; same radio range and bandwidth for all nodes; no node failures. However, since we focus on comparing high-level routing strategies rather than modeling high-fidelity radio behavior, we believe that our relative performance comparisons and discussions are important contributions which are, to the best of our knowledge, underexplored in the literature. Future work could validate these findings in higher fidelity radio models, and multi-round scenarios with energy depletion.

Our results are also limited to the specific parameter configurations we tested (4 node counts, 4 CDR values, 5 speeds, 4 data distributions). While 132,000+ runs provide broad coverage, real-world deployments may involve different parameters, variable radio conditions, and heterogeneous nodes. The custom topologies (3-Shape and inverted-C) represent only two of many possible patterns; other shapes may yield different results. Also, DCE and DCE+ measure data collection performance but do not capture latency, fairness across nodes, or data freshness (Age of Information). Future work should consider multi-objective evaluations, not limited mainly to DCE.

VIII. Conclusions and Future Work

This work investigated how ground mesh network routing strategies affect data collection efficiency when UAV swarms follow fixed trajectories. We proposed the Choreographed Max Flow approach, which pre-computes time-staged data routing schedules using Dinic's maximum flow algorithm, and NODE-GAFT, its energy-aware extension. Our experiments demonstrated that trajectory awareness is the single most important factor for high data collection efficiency, and that specific network topologies with temporally separated contact regions lead to a significant advantage for pre-computed routing (our proposed approaches) over all other approaches we evaluated.

Our experiments across 132,848 simulation runs comparing eight data collection approaches demonstrated that trajectory-aware approaches (MAM: 71.54% mean DCE; Choreographed-MaxFlow: 70.54%; K-Means-FT: 61.22%) achieve substantially higher data collection efficiency than trajectory-agnostic approaches (K-Means: 53.55%; HEED: 56.24%; Liu 2020: 44.57%). All pairwise differences between trajectory-aware and trajectory-agnostic groups are statistically significant ($p < 0.001$). Even simple trajectory-aware adaptations of classical clustering approaches (K-Means-FT, HEED-FT) improve performance by 7–8 percentage points over their trajectory-agnostic counterparts.

In the 128,000 randomized experiments, MAM achieves slightly higher overall DCE than our approach (mean 71.54% vs 70.54%), but this difference is practically negligible ($\hat{A}_{12} = 0.512$). Choreographed-MaxFlow outperforms MAM in 24.62% of paired runs (3,939 out of 16,000 runs), with wins distributed across all CDR levels and individual improvements reaching up to +30.69 percentage points. On energy efficiency (EEff), our approach outperforms MAM in 70.3% of non-tied paired runs ($\hat{A}_{12} = 0.659$, medium effect), demonstrating that pre-computed routing is more energy-efficient than reactive routing in the majority of scenarios.

We designed two custom topologies (3-Shape and inverted-C) with temporally separated contact regions to test our hypothesis that pre-computed routing provides a major advantage when ground-air temporal hiatuses exist. In those scenarios, our approaches Choreographed Max-Flow and NODE-GAFT performed significantly better than other approaches. Our approaches win in 10/15 configurations on the 3-Shape topology (+14.0 pp at speed 0.02, data=500) and 9/15 on the C-Shape topology (+14.6 pp at speed 0.02, data=500). We also conducted an extended validation with 4,608 additional randomized runs (this time following our specified topological patterns) across 64 random topology variants, which supports these findings: for the 3-Shape, Choreographed-MaxFlow wins on 69% of paired runs ($p < 0.001$, $\hat{A}_{12} = 0.690$, medium effect); for the C-Shape, it wins on 53.8% ($p = 0.008$, $\hat{A}_{12} = 0.538$) of the cases. These results indicate that the number of temporally separated contact regions correlates with the advantage of pre-computed routing over reactive routing.

The simple temporal hiatus detector we propose achieves 98.4% accuracy (100% precision, 96.9% recall) in identifying topologies that favor pre-computed routing, enabling topology-adaptive algorithm selection at the moment of ground-air encounter (AMEnc MEET). This enables building a hybrid routing strategy that combines MAM (for general deployments) with Choreographed-MaxFlow (for topologies with temporal hiatuses), using node positions, state and the UAV trajectory as input.

Future work may include: (1) real-world hardware validation using UAV and sensor platforms, addressing the known gap between simulation and real-world results [16]; (2) evaluate the proposed hybrid topology-adaptive strategy that automatically selects between MAM and Choreographed-MaxFlow based on the hiatus detector; (3) conduct multi-round NODE-GAFT experiments where energy depletion across rounds would possi-

DOI: https://doi.org/10.65218/jius.2026.124

bly differentiate it from Choreographed-MaxFlow; (4) conduct broader topology exploration beyond the 3-Shape and inverted-C shapes, including more complex natural formations such as river-following sensor deployments, which may also include different UAV swarm sizes and UAV swarm formation shapes.

Since our experiments have many limitations we discussed, such as its simple radio model without interference, limited air network variation (we evaluated only a 5-UAV V-shape formation), custom topologies limited to two specific shapes, we believe that this work should be seen as a first (exciting) step toward understanding the complex interactions between UAV fixed trajectories constraints, ground network topologies, and routing strategies, rather than a definitive solution. We hope that our findings and proposed approaches will inspire further research and real-world validation in this important area of UAV-WSN data collection.

Acknowledgment

This study was financed in part by the National Council for Scientific and Technological Development (CNPq).
This study was financed in part by AFOSR grant FA8655-25-1-8035.

References

[1] S. Misra, S. C. Misra, and I. Woungang, *Guide to Wireless Mesh Networks*. Springer, 2009.

[2] X. Ma, S. Chisiu, R. Kacimi, and R. Dhaou, "Opportunistic communications in WSN using UAV," in *2017 14th IEEE Annual Consumer Communications and Networking Conference*, 2017, pp. 510–515.

[3] M. T. Nguyen, C. V. Nguyen, H. T. Do, H. T. Hua, T. A. Tran, A. D. Nguyen, G. Ala, and F. Viola, "UAV-assisted data collection in wireless sensor networks: A comprehensive survey," *Electronics*, vol. 10, no. 21, 2021.

[4] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[5] T. Reimer, B. Olivieri, M. Cavalcanti, and M. Endler, "From air to ground: Coordinating UAVs and UGVs in SAR missions," in *2025 28th International Conference on Information Fusion (FUSION)*, 2025, pp. 1–6.

[6] M. P. JV, B. J. O. de Souza, and M. Endler, "Exploring data collection on bluetooth mesh networks," *Ad Hoc Networks*, vol. 130, p. 102809, 2022.

[7] A. Arcuri and L. Briand, "A practical guide for using statistical tests to assess randomized algorithms in software engineering," in *Proceedings of the 33rd International Conference on Software Engineering*, 2011, pp. 1–10.

[8] aiping Pang, F. Chao, H. Zhou, and J. Zhang, "The method of data collection based on multiple mobile nodes for wireless sensor network," *IEEE Access*, vol. 8, pp. 14 704–14 713, 2020.

[9] X. Gao, X. Zhu, and L. Zhai, "AoI-sensitive data collection in multi-UAV-assisted wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5185–5197, 2023.

[10] C. H. Nguyen and K. V. Nguyen, "Multi-UAV assisted data gathering in WSN: A MILP approach for optimizing network lifetime," in *Proceedings – 2021 RIVF International Conference on Computing and Communication Technologies*, 2021.

[11] Y. Wang, Z. Hu, X. Wen, Z. Lu, and J. Miao, "Minimizing data collection time with collaborative UAVs in wireless sensor networks," *IEEE Access*, vol. 8, pp. 98 659–98 669, 2020.

[12] S. Alfattani, W. Jaafar, H. Yanikomeroglu, and A. Yongacoglu, *Multi-UAV Data Collection Framework for Wireless Sensor Networks*. IEEE, 2019.

[13] Y. Liu, J. Yan, and X. Zhao, "Deep-reinforcement-learning-based optimal transmission policies for opportunistic UAV-aided wireless sensor network," *IEEE Internet of Things Journal*, vol. 9, no. 15, pp. 13 823–13 836, 2022.

[14] D. Liu, Y. Xu, Y. Xu, Y. Sun, A. Anpalagan, Q. Wu, and Y. Luo, "Opportunistic data collection in cognitive wireless sensor networks: Air-ground collaborative online planning," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8837–8851, 2020.

[15] Z. Wei, M. Zhu, N. Zhang, L. Wang, Y. Zou, Z. Meng, H. Wu, and Z. Feng, "UAV-assisted data collection for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 460–15 483, 2022.

[16] B. J. O. De Souza, T. Lamenza, M. Paulon, V. B. Rodrigues, V. G. A. Carneiro, and M. Endler, "Collecting sensor data from WSNs on the ground by UAVs: Assessing mismatches from real-world experiments and their corresponding simulations," in *2023 IEEE Symposium on Computers and Communications (ISCC)*, 2023, pp. 284–290.

[17] C. Tunca, S. Isik, M. Y. Donmez, and C. Ersoy, "Distributed mobile sink routing for wireless sensor networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 877–897, 2014.

[18] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks*, vol. 8, no. 1, pp. 7:1–7:31, 2011.

[19] E. Tuyishimire, A. Bagula, and A. Ismail, "Clustered data muling in the internet of things in motion," *Sensors*, vol. 19, no. 3, 2019.

[20] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[21] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.

[22] S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, vol. 5, pp. 4298–4328, 2017.

[23] V. P. Raj and M. Duraipandian, "An energy-efficient cross-layer-based opportunistic routing protocol," *Journal of Engineering Research*, vol. 12, no. 1, pp. 122–132, 2024.

[24] M. Kang and S. W. Jeon, "Energy-efficient data aggregation and collection for multi-UAV-enabled IoT networks," *IEEE Wireless Communications Letters*, vol. 13, no. 4, pp. 1004–1008, 2024.

[25] T. E. Harris and F. S. Ross, "Fundamentals of a method for evaluating rail net capacities," RAND Corporation, Tech. Rep. RM-1573, 1955.

[26] L. R. Ford Jr and D. R. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[27] N. Megiddo, "Optimal flows in networks with multiple sources and sinks," *Mathematical Programming*, vol. 7, no. 1, pp. 97–107, 1974.

[28] E. A. Dinic, "Algorithm for solution of a problem of maximum

DOI: https://doi.org/10.65218/jius.2026.124

flow in networks with power estimation," in *Soviet Math. Doklady*, vol. 11, 1970, pp. 1277–1280.

[29] I. Y. Ibrahim and I. M. Zebari, "A comprehensive review of the Ford–Fulkerson algorithm for network flow problems," *Asian Journal of Research in Computer Science*, vol. 18, no. 3, pp. 335–344, 2025.

[30] A. Buss and A. Al Rowaei, "A comparison of the accuracy of discrete event and discrete time," in *Proceedings of the 2010 Winter Simulation Conference*, 2010, pp. 1468–1477.

[31] A. Vargha and H. D. Delaney, "A critique and improvement of the CL common language effect size statistics of McGraw and Wong," *Journal of Educational and Behavioral Statistics*, vol. 25, no. 2, pp. 101–132, 2000.