CEDAN: Cost-Effective Data Aggregation for UAV-Enabled IoT Networks

Abhishek Bera, *Member, IEEE*, Sudip Misra, *Senior Member, IEEE*, Chandranath Chatterjee, and Shiwen Mao, *Fellow, IEEE*

Abstract—One of the crucial challenges in networked Unmanned Aerial Vehicles (UAVs) is to configure them to serve as aerial base stations (BSs) for collecting data from distributed Internet of Things (IoT) devices in a region devoid of backbone connectivity. To address this challenge, it is required to compute optimized trajectories of UAVs to collect data while considering the different activation patterns of IoT devices. We propose a scheme to optimize the trade-off between the number of covered IoT devices and the travel time of UAVs. The formulated cost minimization problem is the capacitated single depot vehicle routing problem (CSDVRP), which is NP-hard. We propose a solution scheme named Cost-Effective Data Aggregation for UAV-Enabled IoT Networks (CEDAN), which operates in four steps. First, it determines the optimized hovering locations (HLs) for UAVs. Subsequently, CEDAN determines the optimized route adopting Christofides's approximation algorithm for Travelling Salesman Problem (TSP). Further, a split function produces the optimized trajectories for all UAVs. Finally, a route adjustment algorithm applies the cost function and rearranges the order of visiting each HL. Extensive simulation results depict that the CEDAN outperforms than Clarke-Wright (CW) savings heuristics, CEDAN without route adjustment (CWRA), and Zhan *et al.*, respectively.

Index Terms—Unmanned aerial vehicle (UAV), Capacitated single depot vehicle routing problem (CSDVRP), christofides algorithm, IoT networks

1 Introduction

Recently, multi-UAV networks draw attention in terms of aerial BS for several applications areas such as cellular network coverage [1], temporary communications at the disaster area, civilian applications [2], and IoT applications [3], [4], [5]. UAVs are especially useful when there is a lack of infrastructure-based communications or a high probability of communication void. On the other hand, we live in the Internet of Things (IoT) era. The term "Things" represents every sort of sensor or device which are connected to the Internet, including smart environmental monitoring devices, intelligent parking devices, smart grocery tracking sensors, smart home automation systems, and many more [6]. These IoT devices are easy to deploy and low in cost [7]. Therefore, the increasing number of connected devices incurs communication challenges in the limited traditional terrestrial network area. This work explores multiple UAVs in terms of aerial BSs to collect the data from such IoT devices with the flight time constraint. Specially, we determine the optimized trajectories of UAVs to cover the maximum number of IoT devices and minimize the overall travel costs of UAVs. The cost of travel is associated with the travel time of UAVs and the existing active IoT devices at the target locations.

1.1 Motivation

UAVs exhibit immense potential as aerial BSs [8], [9], [10]. Notably, they are useful in emergency communication, limited terrestrial communication, and dense areas where high-rise constructions block traditional cellular communication. UAVs can fly up to a limited altitude, typically in the order of a few tens of meters, and establish strong line-of-sight (LoS) downlink and uplink communications [9]. Therefore, the communication link develops secure and reliable communication with the ground devices [3], [9]. On the other hand, the cost of the UAV is low compared to the terrestrial communication system.

In this work, we study the usage of multiple UAVs to provide aerial communication systems to IoT applications. These UAVs establish secure LoS communication and fly towards the IoT devices to construct a reliable and secure connection [4], [11] to collect the data. Additionally, the communication approach is appropriate for IoT devices as IoT devices are unable to communicate data via long-distance for limited energy [3].

Existing works explore the use of multiple UAVs in IoT applications for data collection. Some of the recent works propose the selection scheme of UAVs [12], the determination of energy-efficient UAV trajectory, the energy optimization of IoT devices [13] to collect the data, minimization of

- *Abhishek Bera is with the Advanced Technology Development Centre, Indian Institute of Technology, Kharagpur 721302, India. E-mail: a.bera.gcect@gmail.com.*
- *Sudip Misra is with the Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. E-mail: smisra@sit.iitkgp.ac.in.*
- *Chandranath Chatterjee is with the Department of Agricultural and Food Engineering, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. E-mail: cchatterjee@agfe.iitkgp.ac.in.*
- *Shiwen Mao is with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA. E-mail: smao@auburn.edu.*

Manuscript received 5 June 2021; revised 2 Feb. 2022; accepted 26 Apr. 2022. Date of publication 5 May 2022; date of current version 4 Aug. 2023.
The work of Shiwen Mao was supported in part by NSF under Grant ECCS-1923163.
(Corresponding author: Sudip Misra.)
Digital Object Identifier no. 10.1109/TMC.2022.3172444

data collection time considering imperfect CSI [14], minimization of AoI to collect the data from IoT devices [15], [16], and maximization of data rates between UAV-BS and user equipment (UE) [17]. On the other hand, few of the current works explore optimal deployment of UAVs [3], determination of optimal trajectory [4], and assignment schemes for IoT devices [18]. However, IoT devices are distributed over a large area and follow different activation patterns [3]. For example, smart temperature monitoring devices transmit data periodically. However, a traffic congestion control IoT device transmits the data when congestion exceeds a threshold point. Therefore, there is a requirement to group distributed IoT devices based on their geographic locations and analyze the activation patterns before deploying UAVs to collect data. Additionally, the optimized trajectories of UAVs are required to cover the maximum active IoT devices. We address these challenges in this paper. Hence, we introduce a scheme, CEDAN, to cover the maximum number of IoT devices and minimize the total travel time of UAVs. Precisely, we aim to establish a fair trade-off between the number of covered IoT devices and the total travel time of UAVs.

1.2 Contribution

We attempt to address three aspects: (a) creating groups of the IoT devices based on their locations and finding optimized hovering locations (HLs) for UAVs to collect the data from IoT devices, (b) designing cost of travel for UAVs considering the limited flight time of UAVs and the activation pattern of IoT devices, and (c) finding an optimized trajectory for each UAV to cover the maximum number of IoT devices, while minimizing the total travel cost for UAVs. To attain these aspects, we introduce CEDAN. The summary of the *contributions* is described below.

- We propose a scheme CEDAN to maximize the number of IoT devices, while minimizing the travel time. We design a cost function considering activation pattern [3] of IoT devices and travel time of UAVs.
- We formulate a minimization problem to minimize the total travel cost. The formulated cost minimization problem is known as CSDVRP, which is NP-hard [19]. Additionally, the feasible solution set of the problem is non-convex due to the presence of a boolean variable [20].
- In order to get the solution to the formulated objective, first, we create the cluster of IoT devices and determine the optimized HLs in line with the work reported by Bera *et al.* [21]. Next, we define the cost of travel for UAVs. Further, we propose an algorithm for CEDAN. The algorithm solves the formulated CSDVRP problem adopting Christofides's [22] approximation algorithm for TSP. Finally, route adjustment algorithm changes the sequence of visiting the HLs for UAVs applying cost function to increase the number of covered IoT devices while minimizing the travel time of UAVs.
- Extensive simulation results describe that CEDAN outperforms the benchmark solutions. Especially, the reliability of the system improves by 43.86%, 50.46%, and 47.82% higher than CW, CWRA, and Zhan *et al.*, respectively.

2 Related Work

Most of the existing works for UAV-enabled data aggregation schemes for IoT devices primarily focused on the energy optimization and optimized deployment of UAVs. In this section, we discuss these two aspects of the existing literature and the gap of the existing research.

2.1 Energy Optimization

Yan *et al.* [12] addressed the selection of BSs and access to bandwidth for UAVs in IoT communication platform. They adopted evolutionary game theory for selecting the BS and Stackelberg game for allocating the BS bandwidth. Motlagh *et al.* [23] proposed a robust system orchestrator to manage multiple UAVs, which have on-board multiple IoT devices to handle IoT events. They offered an energy-aware scheme to minimize the energy consumption, a delay-aware scheme to reduce the operational time, and a scheme to optimize the trade-off between energy consumption and operational time of UAVs. Lin *et al.* [24] addressed a trade-off between system throughput and energy efficiency of IoT devices while UAVs collect the data from IoT devices. They proposed a scheme employing two switching modes of the system as "system-efficiency mode" and "energy-efficiency mode" to enhance the performance of the system. Zhan *et al.* [13] proposed a scheme to minimize the maximum energy consumption of rotary-wing UAV during data collection from IoT devices. They jointly optimized the transmit power for IoT devices, the trajectory of UAV, and scheduling of the communication. However, the entire scheme was designed for a single UAV system. Sun *et al.* [15] emphasized an energy-aware data collection scheme by minimizing the cumulative sum of AoI, transmission energy of IoT devices, and propulsion energy of UAV. Sikeridis *et al.* [25] proposed a framework where IoT devices for public safety networks (PSN) harvests energy from UAV through wireless power communication. Additionally, they introduced utility-based non-orthogonal multiple access (NOMA) transmission power allocation among the IoT devices.

2.2 UAV Deployment

Mozaffari *et al.* [3] investigated an efficient deployment of multiple UAVs to collect data from IoT devices. Authors aimed to optimize the UAV-IoT devices association, uplink communication from IoT devices to UAVs, and 3D placement of multiple UAVs to collect the data from IoT devices. Samir *et al.* [11] aimed to off-load traffic from the traditional network using UAVs to collect data from time-constrained IoT devices. They jointly optimized UAV trajectory and channel allocation to achieve the goal. Bushnaq *et al.* [4] focused on field estimation and data aggregation for IoT devices using UAVs. Especially, they proposed a scheme to minimize the total travel time of a UAV while collecting data from IoT devices. Li *et al.* [18] proposed a joint UAV configuration and IoT node assignment scheme. Notably, they considered long term seamless coverage for IoT devices using multiple UAVs. Yang *et al.* [14] proposed a deep learning-based trajectory planning for a UAV in order to collect the data from IoT nodes. They considered the UAV movement constraints, imperfect CSI for the urban 3D environment, and throughput constraints. However, the proposed work is

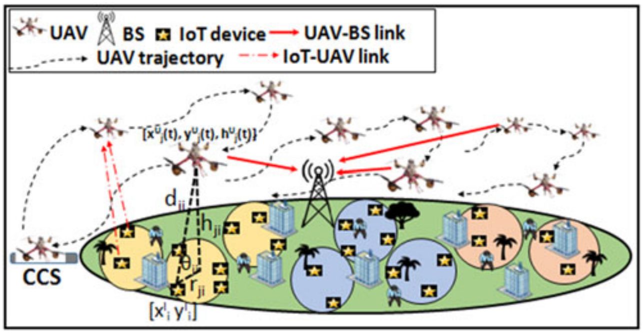


Fig. 1. Network architecture.

restricted to a single UAV system. It is difficult for a single UAV system to cover a large area due to its limited flight time and resources. Zhong *et al.* [17] described a framework where UAVs act as flying BSs in order to maximize the data rates of moving ground UE. Specifically, the authors focused on adjusting the location of UAVs using a deep Q-network to maximize the data rate. However, the analysis to cover a maximum number of devices over a wide area is out of the scope of this work. Tong *et al.* [16] optimized the UAV trajectory in order to collect the data from IoT devices. Specifically, they focused on the minimization of packet loss from IoT devices. However, the work is limited to a single UAV system. Additionally, the analysis to cover the maximum number of IoT devices of a large distributed IoT network scenario was not considered in this work.

Synthesis: Existing solutions aim to optimize trajectories of UAVs, minimize the energy for the uplink transmission from IoT devices to UAVs, minimize the total travel time of UAVs, and efficient deployment of UAVs for collecting data from IoT devices. However, none of the existing works attempt to increase the total number of covered IoT devices considering the limited flight time of UAVs and activation pattern of IoT devices. Especially, there is a requirement to optimize a trade-off between the number of covered IoT devices and the travel cost of UAVs, while considering the activation pattern of IoT devices. In this work, we concentrate on these issues while collecting the data from IoT devices using multi-UAV networks. Therefore, we propose CEDAN to maximize the total number of covered IoT devices while minimizing the total travel costs of UAVs.

3 System Model

We consider that multiple UAVs are deployed in a geographical area to collect data from IoT devices for a time slot $\mathcal{T}$. A set $\mathcal{I} = \{a_1, a_2, a_3, \ldots a_M\}$ consists of M IoT devices that are deployed in a geographical area to monitor the condition of the environment, smart waste management, and smart street light management [3], [4], [11]. On the other hand, a set $\mathcal{U} = \{u_1, u_2, u_3, \ldots u_N\}$ of N UAVs are deployed to collect and transfer data from IoT devices to BS. We assume that UAVs have object detection and collision avoidance ability to have a safe flight for simplicity of this work. Fig. 1 shows the architecture of the proposed work. First, UAVs receive instructions from the BS for the target trajectories. A single trajectory comprises multiple HLs, where a UAV hovers and collects the data from IoT devices during its flight. A set $\mathcal{Z} = \{z_1, z_2, z_3, \ldots z_L\}$ of L such HLs. We consider that the number of HLs is more than the number of deployed UAVs. Therefore, there is a requirement to find out the optimized value of N to cover the HLs. Next, each UAV visits the respective HLs to collect and send the data from IoT devices to BS. Finally, all UAVs return to the central charging station (CCS) for charging, as shown in Fig. 1. Determining trajectory and z_e for each UAV is crucial as a UAV requires covering the maximum number of IoT devices during its flight. We discuss the determination of trajectories and $\mathcal{Z}$ later in this paper. Therefore, the total travel time of a UAV is divided into two parts — a) time to travel and b) time to hover. We assume that $\tau_j^{tra}(z_{e'}, z_e)$ represents the travel time between $z_{e'}$ and z_e ($z_e, z_{e'} \in \mathcal{Z}$), and $\tau_j^{hov}(z_e)$ represents the hovering time at z_e of a u_j for a time slot $\mathcal{T}$. Also, there exists a requirement to create clusters of IoT devices as they are distributed over an area. We assume that $\mathcal{A}_e$ represents a cluster. Therefore, HLs are associated with the clusters, as UAVs collect data from IoT devices at these locations. Additionally, we assume that each UAV has a maximum flight time of τ_{max}. Further, IoT devices establish a connection with UAVs through wireless links. On the other hand, UAVs are connected with the BS using the licensed cellular band. Additionally, we adopt a time division multiple access (TDMA) scheme for IoT devices to transfer the data to an associated UAV. For simplicity, we consider that IoT devices are static, and average speed of each UAV is v_{avg}. The location of each IoT device and each UAV at time instant $t, (t \in [0 - \mathcal{T}])$ are represented as $l_i^I = [x_i^I, y_i^I]$ and $l_j^U(t) = [x_j^U(t), y_j^U(t), h_j^U(t)]$, respectively. We assume that each UAV maintains a considerable height ($h_j^{min} \leq h_j^U(t) \leq h_j^{max}$) that avoids all the obstacles and able to produce a strong line-of-sight connection. Additionally, BS executes the process of determining the trajectories of UAVs and estimating the number of active IoT devices.

3.1 Activation Model of IoT Devices

Based on the application, the activation patterns are different for IoT devices. For example, IoT devices follow a periodic activation model in the case of environmental monitoring, home automation system, and smart grid system [3]. On the other hand, some applications such as intelligent traffic congestion control, smart fire sensing system, and smart health monitoring follow the random activation model [3]. These IoT devices are distributed over an area and follow different activation models. Therefore, there is a requirement for a proper analysis of activation patterns of such IoT devices so that the deployed UAVs can cover the maximum number of active IoT devices. Hence, analysis of the activation model of IoT devices is crucial before deploying the UAVs to the specified area for collecting the data. In line with the work in [3], we adopt two activation models for IoT devices — a) periodic activation, and b) random activation.

In periodic activation such as the environmental monitoring system and smart meter, we assume that the activation pattern is known to the BS in advance, as described in [3]. The BS has the prior information that an IoT device a_i active each δ_i seconds for $\mathcal{T}$. Therefore, we determine the number of activation of a_i in a time slot $\mathcal{T}$ as $[\frac{\mathcal{T}}{\delta_i}]$.

In the random activation model, the IoT devices are activated randomly. The IoT devices are active for a short period and create bursty traffic [3]. As described in the 3 rd generation partnership project (3GPP) [26], and the work in [3], we adopt the beta distribution for the random activation model for a_i. The probability density function of beta distribution for a_i being active at a time instant t ($t \in [0 - \mathcal{T}]$) is expressed as [3], [26]:

$$F(t) = \frac{t^{\alpha-1}(\mathcal{T} - t)^{\beta-1}}{\mathcal{T}^{\alpha+\beta-1}B(\alpha, \beta)}, \tag{1}$$

where α and β are the shape parameters, $\alpha, \beta \geq 0$, and $B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$ represents the beta function using the shape parameters α and β. The BS estimates the number of active devices for a time slot based on the activation patterns. This information is crucial before deploying the UAVs to a specific area. This is because the proposed work aims to cover maximum active IoT devices while minimizing the travel cost of UAVs.

3.2 IoT-UAV Communication Model

UAVs act as flying BSs. Therefore, the probability of LoS signal between a_i and u_j is higher than non-LoS (NLoS) signal. The LoS signal probability is expressed as [8], [9]:

$$Pr_{ij}^{LoS} = \left(1 + \mathbb{X}exp(-\mathbb{Y}[\theta_{ij} - \mathbb{X}])\right)^{-1}, \tag{2}$$

where $\mathbb{X}$ and $\mathbb{Y}$ are the constant values that depend on the environmental condition (urban, rural, dense urban, and other areas), θ_{ij} represents elevation angle between a_i and u_j. Additionally, $\theta_{ij} = (180/\pi)sin^{-1}(h_j^U(t)/d_{ij})$, where $d_{ij} = \sqrt{(x_j^U(t) - x_i^I)^2 + (y_j^U(t) - y_i^I)^2 + (h_j^U(t))^2}$ denotes the distance between a_i and u_j. Also, the probability for NLoS link is $Pr_{ij}^{NLoS} = 1 - Pr_{ij}^{LoS}$. In line with the existing literature (e.g., [8], [9]), the path loss for LoS and NLoS signal between a_i and u_j is expressed as:

$$\begin{aligned} PL_{ij}^{LoS} &= \eta_{LoS}(K_0 d_{ij})^{\gamma}, \\ PL_{ij}^{NLoS} &= \eta_{NLoS}(K_0 d_{ij})^{\gamma}, \end{aligned} \tag{3}$$

where $K_0 = (4\pi f_c)/c$, f_c represents the carrier frequency, c denotes the speed of light, γ represents the path loss exponent, and η_{LoS} and η_{NLoS} are the attenuation factor for LoS and NLoS connections, respectively. We compute the average path loss depending on the LoS and NLoS connections between a_i and u_j as:

$$\begin{aligned} \overline{PL}_{ij} &= \left[(Pr_{ij}^{LoS} PL_{ij}^{LoS}) + (Pr_{ij}^{NLoS} PL_{ij}^{NLoS})\right] \\ &= [A_0 Pr_{ij}^{LoS} + \eta_{LoS}](K_0 d_{ij})^{\gamma}, \end{aligned} \tag{4}$$

where $A_0 = \eta_{LoS} - \eta_{NLoS}$. Therefore, the average channel gain between u_j and a_i at time instant t is $\overline{G}_{ij}(t) = [A_0 Pr_{ij}^{LoS} + \eta_{LoS}]^{-1}(K_0 d_{ij})^{-\gamma}$. We assume that the transmit power of a_i at time instant t is $\mathcal{P}_i(t)$, where $0 < \mathcal{P}_i(t) < P_{max}$ and P_{max} denotes the maximum allowable transmit power for a IoT device. As we mentioned earlier, a UAV employ the TDMA to collect the data from IoT devices. Therefore, we neglect the intra-cluster interference. However, there exist inter-cluster interference due to other UAVs. Hence, the signal-to-interference-plus-noise-ratio (SINR) between u_j and a_i at time instant t is expressed as:

$$\Gamma_{ij}(t) = \frac{\mathcal{P}_i(t)\overline{G}_{ij}(t)}{\sum_{q=1, q\neq j, m\neq i}^{N} P_q(t) G_{mq}(t) + \sigma^2}, \tag{5}$$

where σ^2 denotes the variance of Gaussian noise. Finally, the approximated data transfer rate between a_i and u_j at time instant t using Shannon's theorem is expressed as:

$$\zeta_{ij}(t) = \mathbb{B}\log{}_2[1 + \Gamma_{ij}(t)], \tag{6}$$

where $\mathbb{B}$ represents the channel bandwidth.

3.3 Analysis of the Cost Function

There is a cost associated with the travel of each u_j. In this work, we praise on the two aspects — a) traveling cost of UAVs, and b) total number of IoT devices covered by all the UAVs. Therefore, we model associated travel cost between $z_{e'}$ and z_e for a u_j based on travel time and number of active devices that are ready to transfer the data to the u_j at z_e. We assume that u_j moves from $z_{e'}$ to z_e from time instant t_{n-1} to t_n, t_{n-1}, $t_n \in [0 - \mathcal{T}]$, i.e., u_j moves from $\mathcal{A}_{e'}$ to $\mathcal{A}_e$. As discussed in Section 3.1, we consider the periodic and random activation model for a_i. Therefore, each $\mathcal{A}_e$ consists of a bunch of IoT devices that follow either a random or periodic activation model. Let us assume that a $\mathcal{A}_e$ consists of Y_e^{per} number of IoT devices that follow the periodic activation model. Therefore, $\mathcal{A}_e$ consists of ($Y_e^{rnd} = |\mathcal{A}_e| - Y_e^{per}$) number of IoT devices that follow random activation model. Therefore we express the cost function between $z_{e'}$ and z_e for u_j as:

$$C_j(z_{e'}, z_e) = \left[Y_{z_e}^j(t_n)\left(1 - \frac{\tau_j^{tra}(z_{e'}, z_e)}{\tau_{max}}\right)\right]^{-1}, \tag{7}$$

where $Y_{z_e}^j(t_n) = (\sum_{i=1}^{Y_e^{per}} \phi_i^{per}(t_n) + \sum_{i=1}^{Y_e^{rnd}} \phi_i^{rnd}(t_n))$. $\phi_i^{per}(t_n)$ and $\phi_i^{rnd}(t_n)$ are two boolean variables associated with IoT devices which follow periodic and random activation model, respectively. These two boolean variables are expressed as:

$$\phi_i^{per}(t_n) = \begin{cases} 1, & \text{if ai is active at tn,} \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

$$\phi_i^{rnd}(t_n) = \begin{cases} 1, & \text{if F(tn)≥Fth,} \\ 0, & \text{otherwise,} \end{cases} \tag{9}$$

where F_{th} denotes a predefined threshold. The value of F_{th} may vary with the applications and decides by the BS administrator. Additionally, movement of u_j can be adjusted such a way that δ_i for periodic activation time of IoT devices syncs with t_n. On the other hand, Equation (7) has two primary variables. First, $Y_{z_e}^j(t_n) = (\sum_{i=1}^{Y_e^{per}} \phi_i^{per}(t_n) + \sum_{i=1}^{Y_e^{rnd}} \phi_i^{rnd}(t_n))$ denotes estimated number of active IoT devices at time instant t_n at $\mathcal{A}_e$. Second, $\tau_j^{tra}(z_{e'}, z_e)$ is the travel time for u_j from $z_{e'}$ to z_e. $C_j(z_{e'}, z_e)$ can be reduced if we increase the value of $Y_{z_e}^j(t_n)$ and reduce the value of $\tau_j^{tra}(z_{e'}, z_e)$, as described in Equation (7) that. Hence, the minimization of the cost function leads to an optimized trade-off between the number of covered IoT devices and travel time of UAVs.

3.4 Analysis of Total Travel Time

The total travel time of each UAV consists of the time to hover at associated HLs to collect the data and time from traveling between HLs. Therefore, the total travel time of u_j for $\mathcal{T}$ is expressed as:

$$\tau_j^{tot} = \sum_{z_e, z_{e'} \in \mathcal{Z}} \psi(z_{e'}, z_e)\Big[\tau_j^{tra}(z_{e'}, z_e) + \tau_j^{hov}(z_e)\Big] + \mathbb{T}_j, \tag{10}$$

where the value of boolean variable $\psi(z_{e'}, z_e) \in [0, 1]$ is 1, if a UAV selects a path between $z_{e'}$ and z_e. Otherwise, $\psi(z_{e'}, z_e) = 0$. A UAV hovers at z_e when $\psi(z_e, z_{e'}) = 1$. Additionally, $\mathbb{T}_j \geq 0$ is expressed as $\mathbb{T}_j = \mathbb{T}_j^1 + \mathbb{T}_j^2 + \mathbb{T}_j^3$, where $\mathbb{T}_j^1$ represents the required travel time of u_j from CCS to the first HL, $\mathbb{T}_j^2$ denotes the hovering time at the first HL, and $\mathbb{T}_j^3$ represents required travel time from the last HL to CCS. The time to hover at location z_e is expressed as:

$$\tau_j^{hov}(z_e) = \sum_{i=1}^{Y_e^{per}} \phi_i^{per}\left(\frac{s_i}{\zeta_{ij}(t)}\right) + \sum_{i=1}^{Y_e^{rnd}} \phi_i^{rnd}\left(\frac{s_i}{\zeta_{ij}(t)}\right), \tag{11}$$

where s_i denotes the the size of the data to be transferred from a_i. Also, $\tau_j^{tra}(z_{e'}, z_e)$ is expressed as:

$$\tau_j^{tra}(z_{e'}, z_e) = \frac{D(z_{e'}, z_e)}{v_{avg}}, \tag{12}$$

where $D(z_{e'}, z_e)$ represents the horizontal distance between $z_{e'}$ and z_e.

3.5 Problem Formulation

This work aims to cover the maximum number of IoT devices while minimizing the overall travel time of UAVs. Therefore, we investigate a solution to establish a trade-off between these two objectives. The cost function comprises the travel time of u_j from $z_{e'}$ to z_e and the estimated number of active IoT devices once the u_j reaches to z_e, as discuss in Section 3.3.

Therefore, we formulate a minimization problem as:

$$\mathbf{P1}: \quad \underset{N, l_j^U(t), \psi(z_{e'}, z_e)}{\text{minimize}} \sum_{j=1}^{N} \sum_{z_e, z_{e'} \in \mathcal{Z}} \psi(z_{e'}, z_e) C_j(z_{e'}, z_e), \tag{13a}$$

$$\text{subject to} \quad \tau_j^{tot} \leq \tau_{max}, \forall j \in [1 - N], \tag{13b}$$

$$\sum_{e=1, e \neq e'}^{L} \psi(z_{e'}, z_e) = 1, \tag{13c}$$

$$\sum_{e'=1, e' \neq e}^{L} \psi(z_{e'}, z_e) = 1, \tag{13d}$$

$$0 < \mathcal{P}_i(t) < P_{max}, \forall i \in [1 - M], \tag{13e}$$

$$\varphi_{z_{e'}} - \varphi_{z_e} + N\psi(z_{e'}, z_e) \leq N - 1, \tag{13f}$$

$$2 \leq z_{e'} \neq z_e \leq N \tag{13g}$$

where N represents the number of UAVs, $l_j^U(t)$ denotes the location coordinate of u_j at time instant t, and the boolean variable $\psi(z_{e'}, z_e) \in [0, 1]$ is 1, if a UAV selects a path between $z_{e'}$ and z_e. Otherwise, $\psi(z_{e'}, z_e) = 0$. Equation (13b) represents the capacity constraints in terms of flight time for u_j, $\forall u_j \in \mathcal{U}$. Further, Equations (13c) and (13d) ensure that there exists only one departure and one arrival of a u_j at each HL, respectively. Equation (13e) represents transmit power constraint of each a_i. Finally, Equations (13f) and (13g) eliminates all sub-routes and ensure the unique routes for all UAVs. The Equations (13f) and (13g) are known as Miller-Tucker-Zemlin equation [4], [27].

4 Problem Analysis and Solution Approach

4.1 Analysis of P1

The objective of CSDVRP proposed by Dantzig in 1959 [28] is to minimize the total cost such that: a) multiple vehicles start from a single depot and end at the same depot, b) the location of each customer is visited once, and c) the capacity and distance restrictions of all the vehicles are preserved. Additionally, the CSDVRP problem is a generalized version of TSP with capacity constraints of vehicles, as it consists of multiple TSPs [28]. On the other hand, in problem P1, Equations (13c) and (13d) ensure that there exists a single departure from each HL, and each HL is visited once, respectively. The Miller-Tucker-Zemlin [4], [27] formulation as described in Equations (13f) and (13g) ensure unique trajectories of all UAVs and eliminate all sub-routes. The objective of P1 is to minimize the total cost. Additionally, Equation (13b) represents capacity constraint in terms of flight time of each u_j. Therefore, we conclude that problem P1 is a CSDVRP problem [28]. The time complexity of VRP and TSP is NP-hard, which is proved by Karp in 1972 [19]. Hence, problem P1 is NP-hard. Additionally, feasible solution sets for P1 is nonconvex due to the presence of the boolean variable [20].

It is a challenging task to solve an NP-hard problem. Existing commercial solvers such as CPLEX, Google Or-Tools, and Gurobi can solve the CSDVRP problem. However, existing solvers do not consider the activation pattern of IoT devices. Therefore, these solvers are not suitable to optimize the trade-off between the total number of covered IoT devices and the travel time of UAVs. Hence, we adopt a heuristic approach to address P1. First, we determine the optimal HLs for UAVs, as IoT devices are distributed over a specified area. After that, we construct a solution that helps to minimize the cost of problem P1. Hence, the total number of covered IoT devices by all UAVs increases.

4.2 Hovering Location Selection

IoT devices are distributed over a geographical area. Therefore, there is a requirement to create clusters based on their geographic coordinates before deploying UAVs for collecting data. Additionally, we determine a suitable HL for each cluster, considering the power constraint of each a_i. We can conclude from the Equations (3) and (5) that the transmit power of each a_i primarily depends on d_{ij}. Therefore, the minimization of total distances from UAVs helps to find appropriate HLs and leads to the solution of P1. Hence, P2 refers to determining the optimal HLs, an initial solution step for problem P1 (as discussed in Section 4.1). Therefore, we can write a minimization problem to find the optimized location of each z_e as:

$$\mathbf{P2}: \quad \underset{l_j^U(t)}{\text{minimize}} \quad \sum_{m=1}^{|\mathcal{A}_e|} d_{jm},$$

$$\text{subject to} \tag{14}$$

$$x_j^{min} \leq x_j^U(t) \leq x_j^{max}, \tag{15}$$

$$y_j^{min} \leq y_j^U(t) \leq y_j^{max}, \tag{16}$$

$$h_j^{min} \leq h_j^U(t) \leq h_j^{max} \tag{17}$$

We previously solved a similar problem as P2 in [21]. In line with [21], we create the cluster applying the K-means algorithm. The centroid of each cluster is the HL for each UAV with a height of $h_j^U(t)$. This is due to the fact that K-mean algorithm ensures the sum of the distances from all IoT devices to the centroid of each cluster is minimum [21]. Therefore, we have the optimal 2D coordinates of each z_e as $[x_e^*, y_e^*]$ using the K-means clustering, as described in [21].

Algorithm 1. Algorithm for Finding Optimized Routes

```
INPUTS:
 1: $\mathcal{G}, \mathcal{Z}, \tau_{max}, \mathcal{I}, \mathcal{U}, \mathbb{C}$
OUTPUT:
 1: $\mathcal{G}'$
PROCEDURE:
 1: function CHRISTOFIDES($\mathcal{G}$, $\mathbb{C}$)
    1) Construct minimum spanning tree $T_{MST}$ from $\mathcal{G}$
    2) Determine the set of odd degree vertices $\mathbb{O}$ from $T_{MST}$
    3) Find minimum-weight perfect matching using $\mathbb{O}$
    4) Connect the edges of $T_{MST}$ and $\mathbb{O}$ and create a Eulerian
       circuit $\mathcal{G}_{EULER}$
    5) Finally, create a Hamiltonian circuit $\mathcal{G}_H$ by removing the
       duplicate vertices from $\mathcal{G}_{EULER}$
2:    return$\mathcal{G}_H$
3: end Function
4: function SPLIT($\mathcal{G}_H$, $\mathbb{C}$)
5:    Split $\mathcal{G}_H$ into multiple sub routes ($\mathbb{R}_j$) such that:
    1) start and end point of each $\mathbb{R}_j$ is CCS
    2) $\tau(\mathbb{R}_j) \leq \tau_{max} - \sum_{z_e \in \mathbb{R}_j, z_e \neq CCS} \tau_j^{hov}(z_e) \triangleright \tau(\mathbb{R}_j)$ is the tot-
       al travel time for $\mathbb{R}_j$
    3) a vertex is a member of only one sub route except the CCS
6:    $N$= total number of sub routes
7:    $\mathcal{G}' \leftarrow (\mathbb{R}_j, u_j), \forall j \in [1 - N]$
8:    return$\mathcal{G}'$
9: end Function
10: Call CHRISTOFIDES($\mathcal{G}$, $\mathbb{C}$)
11: Call SPLIT($\mathcal{G}_H$, $\mathbb{C}$)
12: return $\mathcal{G}'$
```

4.3 Proposed Algorithm

4.3.1 Finding Optimized Routes

We strive to determine the optimized routes to cover IoT devices using multiple UAVs. We follow two steps before searching for the optimized routes. These steps are—a) determining the set of HLs for UAVs and clusters of IoT devices, and b) finding the optimized coordinate for each HL. We already discussed the process for (a) and (b) in Section 4.2. Next, we search the optimized number of UAVs and their trajectories to cover the maximum number of IoT devices while minimizing travel time. To achieve the goal, first, we propose a search algorithm based on Christofides's approximation algorithm [22] to find the optimized routes. We consider a weighted complete connected graph $\mathcal{G} = (\mathcal{V}, E)$, where $\mathcal{V}$ represents the set of vertices and expresses as $\mathcal{V} = \mathcal{Z} \cup \{CCS\}$, E represents the set of all the edges, and the weight of edges is the travel time of a u_i between two vertices. $\mathbb{C}$ represents the adjacent cost matrix for all the edges.

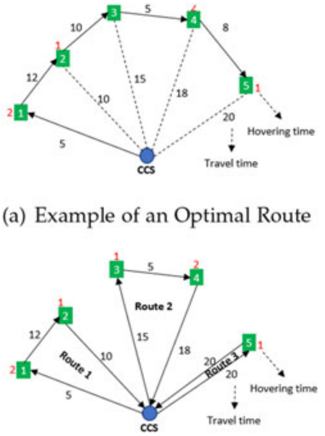
Algorithm 1 incorporates Christofides's approximation algorithm [22] to find the optimized routes for UAVs. In Algorithm 1, the function CHRISTOFIDES($\mathcal{G}, \mathbb{C}$) is based on Christofides's [22] approximation solution for TSP. This function provides the best possible route $\mathcal{G}_H$ using minimum spanning tree, perfect matching, Euler circuit, and Hamiltonian circuit of graph theory, as described in Algorithm 1. We adopt Christofides's approximation algorithm because it ensures that the performance ratio with the optimal solution is no longer than 1.5 [22], which is one of the best among the existing constructive heuristic solutions of TSP [29]. Next, SPLIT($\mathcal{G}_H$, $\mathbb{C}$) function splits the $\mathcal{G}_H$ such that the multiple optimized routes are determined considering the flight time constraints of each UAV. Finally, the Algorithm 1 returns the optimized routes and assigned UAV mapping ($\mathcal{G}'$). It is noteworthy that we determine the optimized number of UAVs that are required to cover the specified area from Algorithm 1. On the other hand, we generate the best possible routes and optimized number of UAVs based on the travel time using Algorithm 1. However, we aim to maximize the number of covered IoT devices and minimize minimum travel time. Therefore, we propose an algorithm to rearrange each route using the cost function as described in Equation (7). The route adjustment algorithm provides a fair trade-off between the number of covered IoT devices and the travel time of UAVs, i.e., the objective of P1.

Algorithm 2. Algorithm for Route Adjustment

```
INPUTS:
 1: $\mathcal{G}', N$
OUTPUT:
 2: $\mathcal{G}^{Final}$
PROCEDURE:
 1: function RADJUSTMENT($\mathcal{G}'$, $N$)
 2:    Create the cost for all the edges that are exist in $\mathcal{G}'$ using
       Equation (7).
 3:    for all $\mathbb{R}_j \in \mathcal{G}'$ do
 4:    $Visited = NULL$.
 5:    Starting from CCS. Mark it as $Source$.
       $Visited = \{Visited, Source\}$.
 6:    Find the nearest vertex which has minimum cost from the
       $Source$ and unvisited vertex.
 7:    Make the nearest vertex as $Source$ and mark it as visited.
 8:    If all the vertices of the route is visited then exit.
       Otherwise, go to step 6.
 9:    Update the route in $\mathcal{G}'$ with the help of $Visited$.
10: end for
11: return $\mathcal{G}^{Final} = \mathcal{G}'$
12: end Function
```

4.3.2 Complexity and Numerical Example of Algorithm 1

Initially, Algorithm 1 adopts Christofides's approximation to generate the Hamiltonian circuit $\mathcal{G}_H$ (see Algorithm 1). Then, it calls a Split function to generate multiple routes for the UAVs. As described and proved in [22], in a worst-case scenario, Christofides's approximation algorithm solves the TSP in polynomial time, which is $O(|\mathcal{Z}|^3)$ [22]. On the other hand, the SPLIT starts from the starting location and considers each location from the $\mathcal{G}_H$ list sequentially. However, it continuously checks if the next location is added to the current list. Then, it evaluates if the UAV can reach from the next location to the start location or not, considering the



(a) Example of an Optimal Route

(b) Split Routes

Fig. 2. Numerical example of algorithm 1.

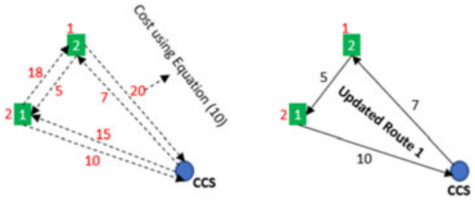
remaining flight time. If yes, then it adds the next location to the current list. Otherwise, it creates a new route starting from the start location. This process continues until it considers all the locations from $\mathcal{G}_H$. Therefore, the time complexity of the SPLIT function is $O(|\mathcal{Z}|)$. Next, we describe the Algorithm 1 with a numerical example as shown in Fig. 2. Suppose, the Christofides's approximation of Algorithm 1 generates a $\mathcal{G}_H$ for a specific instance as shown in Fig. 2a. The weight of the edges represent the travel time, and the value at the node (in red colour) represents the hovering time in Fig. 2a. Additionally, the flight time constraint for this example is 45. Therefore, the Algorithm 1 calls the SPLIT function to generate an optimal number of routes, which is 3 in this example (see Fig. 2b), considering the flight time constraint of each UAV. Hence, the optimal routes for this example case is $\{CCS->1->2->CCS\}$, $\{CCS->3->4->CCS\}$, $\{CCS->5->CCS\}$, as shown in Fig. 2b.

4.3.3 Route Adjustment

Algorithm 2 rearranges the routes using the cost function as described in Equation (7). Function RADJUSTMENT($\mathcal{G}'$, N) adopts a greedy approach to rearrange the routes. For each route, the algorithm starts from CCS. First, it marks all the vertices as unvisited. Thereafter, it makes the vertex as the current vertex and marks it as visited. Then it searches the nearest vertex, which has the minimum cost to visit among the unvisited vertices. Next, it marks the nearest vertex as current and visited. The loop continues until all the vertex are visited for all the routes. Finally, it returns the updated routes.

4.3.4 Time Complexity and Numerical Example of Algorithm 2

Algorithm 2 executes the search for each route and each route executes search operation $|\mathbb{R}_j|(|\mathbb{R}_j|-1)$ times. Therefore, the total number executions is for $\sum_{j=1}^{N} j|\mathbb{R}_j|(|\mathbb{R}_j|-1)$ times. Hence, the time complexity of the Algorithm 2 is $O(|Z|^3)$. On the other hand, we continue the same example that we described in Section 4.3.2. In this example, we describe the Algorithm 2 using one route, Route 1, as shown in Fig. 3. First, Algorithm 2 updates the costs between two vertices using Equation (7), as described in Fig. 3a. Then, it adopts a greedy approach as describe in Algorithm 2. As described in Fig. 3a, the new optimal route for Route 1 is $\{CCS-->2-->1-->CCS\}$ after accumulating the new cost. Finally, Fig. 3b shows the updated route for Route 1. It is notable that the total time of the travel remain same as Algorithm 2 rearranges the sequence of visiting the HLs to increase the total number of active IoT devices.



(a) Updating the Cost for Route 1

(b) Updated Route 1

Fig. 3. Numerical example of algorithm 2.

5 Performance Evaluation

5.1 Simulation Parameters

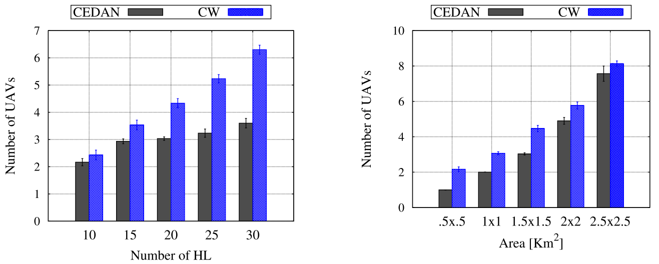
In the simulation, we consider 500 IoT devices are distributed over $1500 \times 1500\ m^2$ area. The simulation parameters are described in the Table 1. The IoT devices are static, and the BS knows the location information of all IoT devices. We consider an urban environment for this work. Therefore, we choose the attenuation factors $\eta_{LoS} = 1.6$ dB, $\eta_{LoS} = 23$ dB [21], path loss exponent $\gamma = 2$ [3], and environment constants $\mathbb{X} = 10.39$ and $\mathbb{Y} = 0.05$ [30] which deal with the urban environment at carrier frequency $f_c = 2$ GHz. Additionally, we consider $P_{max} = 200$ mW, $\alpha = 3$, and $\beta = 4$, according to the existing work in [3]. We consider the activation pattern of each IoT device during simulation, as described in Section 3.1. Additionally, we vary three parameters during simulation to analyze the performance metrics. The parameters are - a) count of HLs, b) area size, and c) P_{max}.

5.2 Performance Metrics

i) *Total travel time:* The total travel time of UAVs is one of the key factors in multi-UAV networks due to the limited flight time of UAVs. Therefore, the primary objective of a scheme is to minimize the total travel time of UAVs for $\mathcal{T}$. The total travel time is expressed for all

TABLE 1
Simulation Parameters

Parameters	Values
Simulation area	$1500 \times 1500\ m^2$
M	500
η_{LoS}	1.6 dB [21]
η_{NLoS}	23 dB [21]
γ	2 [3]
P_{max}	200 mW[3]
σ^2	−95 dBm [21]
f_c	2 GHz [3]
α	3 [3]
β	4 [3]
$\mathbb{X}$	10.39 [30]
$\mathbb{Y}$	0.05 [30]



(a) Optimized number of UAVs vs number of HLs with Area = 1.5x1.5 [Km^2] and P_{max} = 0.2 [W]

(b) Optimized number of UAVs vs number of HLs with $|\mathcal{Z}|$ = 20 and P_{max} = 0.2 [W]

Fig. 4. Optimized number of UAVs.

the UAVs as $T_{tot} = \sum_{j=1}^{N} \tau_j^{tot}$. We vary the number of HLs and the area size to measure the total travel time.

ii) *Total number of covered IoT devices:* One of the primary objectives of this work is to maximize the total number of covered active IoT devices for $\mathcal{T}$. Therefore, the total number of covered IoT devices ($\mathcal{Y}$) indicates the performance of the system. The $\mathcal{Y}$ is computed as $\mathcal{Y} = [\sum_{j=1}^{N} \sum_{z_e \in \mathcal{Z}} Y_{z_e}^j]$. We vary the number of HLs and area size to determine the $\mathcal{Y}$.

iii) *Optimized number of UAVs (N):* An optimized number of UAVs refers to the required number of UAVs to cover the area and collect the data from IoT devices. This parameter indicates the utilization of the resources (UAVs). One of the outcomes of the proposed algorithm, CEDAN, is the number of UAVs that cover the area and collect the data from IoT devices. We vary the number of HLs and the size of the area to calculate the number of UAVs.

iv) *Efficiency (χ):* The system efficiency is calculated from the number of covered IoT devices per unit time for a time slot. The χ is expressed as $\chi = \mathcal{Y}/T_{tot}$. We conduct two simulations varying the number of HLs, and the maximum transmit power of IoT devices to generate the χ. Especially, χ sketches a fair trade-off between $\mathcal{Y}$ and T_{tot}.

v) *Reliability:* The reliability of a system defines the ability of the system to perform under constraints. This work concentrates on covering maximum active IoT devices for $\mathcal{T}$. Therefore, we measure the reliability of the proposed system as a ratio between $\mathcal{Y}$ and the active IoT device count for $\mathcal{T}$. We vary the number of HLs and the size of the area to measure the reliability of the system.

5.3 Benchmarks

As benchmarks, we select the work proposed by Zhan *et al.* as described in [13], Clarke-Wright (CW) savings heuristics [31] algorithm, and CEDAN without the route adjustment (CWRA) to analyze the performance of CEDAN. Zhan *et al*. proposed a scheme to collect the data from IoT devices while optimizing the trajectory of a UAV. They jointly optimized the transmit power for IoT devices, the trajectory of UAV, and the communication schedule. Specifically, the authors formulated an energy minimization problem, considering UAV's flight time constraint and the data collection necessity. They adopted CVX solver, Karush-KuhnTucker (KKT), and proposed an optimization algorithm to solve the problem. Therefore, this work is suitable to compare with our proposed work. CW savings algorithm is one of the widely used solutions for VRP and was proposed by Clarke-Wright in 1964. The algorithm is designed based on the pairwise savings (in terms of cost) between two locations. In this algorithm, the pairwise savings are sorted in decreasing order. After that, it creates multiple Hamiltonian circuits as multiple routes for multiple vehicles considering minimum costs. Therefore, the CW algorithm is suited to compare with CEDAN. On the other hand, CWRA is the simplified version of the proposed scheme CEDAN. In CWRA, we do not use the route adjustment algorithm to emphasize the importance of cost function as described in Equation (7). The comparison between CEDAN and CWRA portrays the trade-off between the number of covered IoT devices and the travel time of UAVs.

5.4 Results and Discussions

5.4.1 Optimized Number of UAVs

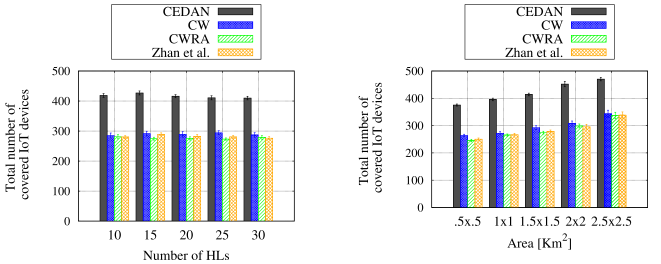
Fig. 4a shows that the optimized number of required UAVs to cover the specified area while varying the count of HLs. The result in Fig. 4a illustrates that the average number of required UAVs in CEDAN is 31.45% less than the CW. On the other hand, Fig. 4b depicts the optimized number of required UAVs while we vary the size of the area. It shows that the average number of required UAVs in CEDAN is 21.61% less than the CW. CEDAN performs better because it adopts Christofides's approximation algorithm, which generates the route (solution for TSP) using a minimum spanning tree. On the other hand, CW follows savings based greedy approach, which is not always optimized.

5.4.2 Total Number of Covered IoT Devices

Fig. 5a sketches the total number of covered IoT devices by all the UAVs for a time slot. In this simulation, we vary the number of HLs. The simulation result shows that $\mathcal{Y}$ in CEDAN is 43.86%, 50.46%, and 47.84% higher than that in CW, CWRA, and Zhan *et al.*, respectively. However, result in Fig. 5b describes that the $\mathcal{Y}$ in CEDAN is 42.44%, 48.31%, and 47.38% higher than that in CW, CWRA, and Zhan *et al.*, respectively. Additionally, we vary the size of the area to generate the result, as shown in Fig. 5b. CEDAN outperforms the CW, CWRA, and Zhan *et al.* because we adopt a route adjustment algorithm employing the modified cost as describe in Equation (7). The cost in Equation (7) considers the number of active IoT devices at the target HL and the travel time between two HLs. Therefore, the UAVs able to cover more active IoT devices.

5.4.3 Reliability

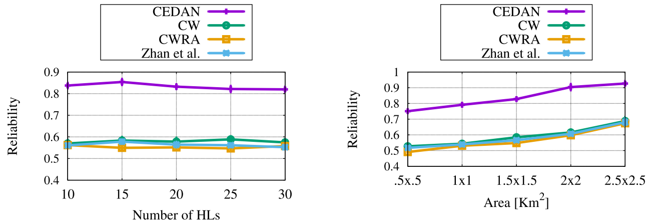
Fig. 6a describes the reliability of the system while varying the count of HLs. The result in Fig. 6a shows that the reliability in CEDAN is 43.86%, 50.46%, and 47.82% higher than CW, CWRA, and Zhan *et al.*, respectively. Also, Fig. 6b depicts that the reliability of the system in CEDAN is 41.98%, 47.83%, and 44.32% higher than CW, CWRA, and Zhan *et al.*, respectively. We vary the size of the area to generate the result, as shown in Fig. 6b. CEDAN performs better than benchmarks because it considers the activation pattern of IoT devices. Further, the cost function in Equation (7) helps route adjustment algorithm to rearrange the



(a) Total number of covered IoT devices vs Number of HLs with area = 1.5x1.5 $[Km^2]$ and P_{max} = 0.2 [W]

(b) Total number of covered IoT devices vs area with $|\mathcal{Z}| = 20$ and P_{max} = 0.2 [W]

Fig. 5. Total number of covered IoT.



(a) Reliability vs number of HLs with area = 1.5x1.5 $[Km^2]$ and P_{max} = 0.2 [W]

(b) Reliability vs area with $|\mathcal{Z}|$ = 20 and P_{max} = 0.2 [W]

Fig. 6. Reliability.

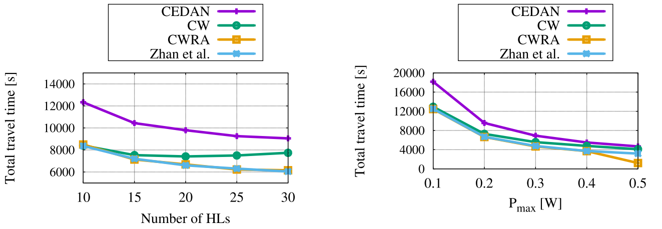
order of HLs. Hence, the number of covered IoT devices is increased.

5.4.4 Total Travel Time

Fig. 7 sketches the total travel time of UAVs to cover the IoT devices. We vary the number of HLs, as shown in Fig. 7a, and the P_{max} of each a_i, as shown in Fig. 7b, to generate the total travel time of UAVs. Results in Fig. 7a describe that total travel time in CEDAN is 24.15%, 31.79%, and 32.14% greater than CW, CWRA, and Zhan *et al.*, respectively. On the other hand, results in Fig. 7b shows that total travel time in CEDAN is 26.93%, 40.08%, and 38.64% higher than CW, CWRA, and Zhan *et al.*, respectively. Fig. 7 shows that the total travel in CEDAN is higher than that in CW, CWRA, and Zhan *et al.* This work aims to cover the maximized number of IoT devices with the minimized travel time of UAVs. Therefore, we apply the route adjustment algorithm following the activation pattern of IoT devices in CEDAN. Hence, the order of visiting the HLs are rearranged, which leads to higher travel time in CEDAN than CW, CWRA, and Zhan *et al.* to cover the higher number of IoT devices, as shown in Section 5.4.2. Therefore, CEDAN provides a fair trade-off between the travel time of UAVs and the number of covered IoT devices (see Fig. 5).

5.4.5 Efficiency

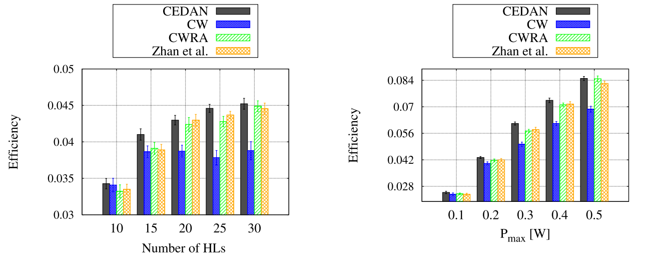
Fig. 8a shows the efficiency of the system while varying the number of HLs. Especially, the result in Fig. 8a describes that the efficiency in CEDAN is 10.64%, 2.8%, and 2.2% higher than that in CW, CWRA, and Zhan *et al.*, respectively. On the other hand, we vary the maximum transmit power of each IoT device to determine the efficiency of the system, as shown in Fig. 8b. Fig. 8b depicts that the efficiency in CEDAN is 17.82%, 3.15%, and 3.6% higher than CW, CWRA, and Zhan *et al.*, respectively. Fig. 8 shows that the efficiency is almost similar in CEDAN compare to the benchmark solution while the reliability and number of covered IoT devices are high in CEDAN as shown in Figs. 6 and 5, respectively.



(a) Total travel time vs number of HLs with area = 1.5x1.5 $[Km^2]$ and P_{max} = 0.2 [W]

(b) Total travel time vs P_{max} with area = 1.5x1.5 $[Km^2]$ and $|\mathcal{Z}| = 20$

Fig. 7. Total travel time.



(a) Efficiency vs number of HLs with area = 1.5x1.5 $[Km^2]$ and P_{max} = 0.2 [W]

(b) Efficiency vs P_{max} with area = 1.5x1.5 $[Km^2]$ and $|\mathcal{Z}| = 20$

Fig. 8. Efficiency.

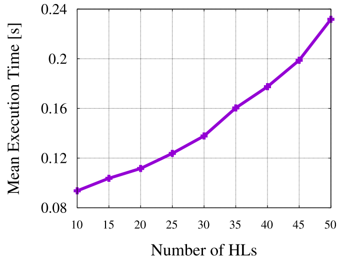


Fig. 9. Execution Time vs Number of HLs with area = 1.5x1.5 $[Km^2]$ and P_{max} = 0.2 [W].

5.4.6 Execution Time

We conducted the simulation for the proposed scheme CEDAN in a computer that has Intel Core i7-10875H 2.3 GHz processor, 32 GB DDR4-SDRAM, Windows 10 as operating system, and 1 TB hard disk. Fig. 9 describe the execution time of the proposed scheme. The result in Fig. 9 describes that the execution time is increased while increasing the number of HLs. The reason is that the Algorithm 1 and Algorithm 2 deals with more number of HLs and routes to generate the optimal routes.

6 Conclusion

In this work, we proposed a novel scheme, CEDAN, to achieve optimized trajectories of multiple UAVs to maximize the number of covered IoT devices. We considered the limited flight time of UAVs and the limited transmit power of IoT devices. We attempted to minimize the total travel cost of UAVs and maximize the number of IoT devices covered by each UAV. To achieve these two goals, we formulated problem P1. P1 is a CSDVRP problem, which is NP-hard. Additionally, the feasible solution set for P1 is non-convex due to the presence of the boolean variable. Therefore, we proposed a heuristic approach for CEDAN. As a solution, first, we created clusters of IoT devices and determined the optimized HLs. Furthermore, we modeled the cost function for a UAV to travel between two HLs, considering the travel time and the number of active IoT devices. Finally, we determined the optimized UAV trajectories with minimized associated costs to collect data from IoT devices. The simulation results illustrated that CEDAN outperforms the CW, CWRA, and Zhan *et al.* Notably, the total number of covered IoT devices are improved by 43.86%, 50.46%, and 47.84% in CEDAN than CW, CWRA, and Zhan *et al.*, respectively.

We intend to optimize the energy efficiency for efficient communication between IoT devices and UAVs and bandwidth allocation in the future. Also, we will analyze collision avoidance techniques for UAVs in future works.

References

[1] X. Zhang and L. Duan, "Fast deployment of UAV networks for optimal wireless coverage," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 588–601, Mar. 2019.

[2] N. Pathak, S. Misra, A. Mukherjee, A. Roy, and A. Zomaya, "UAV virtualization for enabling heterogeneous and persistent UAV-as-a-Service," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6 pp. 6731–6738, Jun. 2020, doi: 10.1109/TVT.2020.2985913.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient Internet of Things communications," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7574–7589, Nov. 2017.

[4] O. M. Bushnaq, A. Celik, H. Elsawy, M. Alouini, and T. Y. Al-Naffouri, "Aeronautical data aggregation and field estimation in IoT networks: Hovering and traveling time dilemma of UAVs," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4620–4635, Oct. 2019.

[5] A. Mukherjee, S. Misra, V. S. P. Chandra, and M. S. Obaidat, "Resource-optimized multiarmed bandit-based offload path selection in edge UAV swarms," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4889–4896, Mar. 2019.

[6] O. B. Sezer, E. Dogdu, and A. M. Ozbayoglu, "Context-aware computing, learning, and Big Data in Internet of Things: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 1–27, Feb. 2018.

[7] A. Brogi and S. Forti, "QoS-Aware deployment of IoT applications through the fog," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1185–1192, Oct. 2017.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, Jun. 2016.

[9] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.

[10] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–5.

[11] M. Samir, S. Sharafeddine, C. Assi, T. Nguyen, and A. Ghrayeb, "UAV trajectory planning for data collection from time-constrained IoT devices," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, Jan. 2020.

[12] S. Yan, M. Peng, and X. Cao, "A game theory approach for joint access selection and resource allocation in UAV assisted IoT communication networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1663–1674, Apr. 2019.

[13] C. Zhan and H. Lai, "Energy minimization in internet-of-things system based on rotary-wing UAV," *IEEE Wireless Commun. Lett.*, vol. 8, no. 5, pp. 1341–1344, Oct. 2019.

[14] Y. Wang *et al.*, "Trajectory design for UAV-based internet-of-things data collection: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 9, no. 5 pp. 3899–3912, Mar. 2021.

[15] M. Sun, X. Xu, X. Qin, and P. Zhang, "AoI-energy-aware UAV-assisted data collection for IoT networks: A deep reinforcement learning method," *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17 275–17 289, Dec. 2021.

[16] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, "Deep reinforcement learning for efficient data collection in UAV-aided Internet of Things," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2020, pp. 1–6.

[17] X. Zhong, Y. Huo, X. Dong, and Z. Liang, "Deep Q-network based dynamic movement strategy in a UAV-assisted network," in *Proc. IEEE 92nd Veh. Technol. Conf.*, 2020, pp. 1–6.

[18] X. Li, H. Yao, J. Wang, S. Wu, C. Jiang, and Y. Qian, "Rechargeable Multi-UAV aided seamless coverage for QoS-Guaranteed IoT networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 902–10 914, Dec. 2019.

[19] R. Karp, *Reducibility Among Combinatorial Problems*. New York, NY, USA: Plenum Press, 1972, pp. 85–103.

[20] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.

[21] A. Bera, S. Misra, and C. Chatterjee, "QoE analysis in cache-enabled Multi-UAV networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6680–6687, Jun. 2020, doi: 10.1109/TVT.2020.2985933.

[22] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem" Graduate Sch. Ind. Admin., Carnegie Mellon Univ., Tech. Rep. 388, 1976.

[23] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-Based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.

[24] X. Lin, G. Su, B. Chen, H. Wang, and M. Dai, "Striking a balance between system throughput and energy efficiency for UAV-IoT systems," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10 519–10 533, Dec. 2019.

[25] D. Sikeridis, E. E. Tsiropoulou, M. Devetsikiotis, and S. Papavassiliou, "Wireless powered public safety IoT: A UAV-assisted adaptive-learning approach towards energy efficiency," *J. Netw. Comput. Appl.*, vol. 123, pp. 69–79, 2018.

[26] "3GPP active work programme," Accessed: Mar. 2020. [Online]. Available: https://www.3gpp.org/DynaReport/FeatureOrStudyItemFile-450015.htm

[27] C. E. Miller, A. W. Tucker, and R. A. Zemlin, "Integer programming formulation of traveling salesman problems," *J. ACM*, vol. 7, no. 4, pp. 326–329, Oct. 1960. [Online]. Available: https://doi.org/10.1145/321043.321046

[28] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, 1959. [Online]. Available: https://EconPapers.repec.org/RePEc:inm:ormnsc:v:6:y:1959:i:1:p:80–91

[29] S. D. Johnson and A.L. McGeoch, *The Traveling Salesman Problem: A Case Study in Local Optimization*, E. H. L. Aarts and J. K. Lenstra, Eds., Chichester, U.K.: Wiley, 1997.

[30] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Glob. Commun. Conf.*, 2014, pp. 2898–2904.

[31] G. Clarke and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations Res.*, vol. 12, no. 4, pp. 568–581, 1964. [Online]. Available: http://www.jstor.org/stable/167703

Abhishek Bera (Member, IEEE) received the BTech degree from the West Bengal University of Technology, in 2009, the ME degree from Bengal Engineering and Science University Shibpur, in 2011, and the PhD degree from the Advanced Technology Development Centre, Indian Institute of Technology Kharagpur, India. He is currently a postdoctoral researcher with the University of Luxembourg. His current research interests include multi-UAV networks, Internet of Things, aerial robotics, 5G, and mobile edge computing. His research works are published in different reputed SCI journals (including IEEE Transactions) and many reputed conferences. He is a reviewer for several international journals and conferences.

Sudip Misra (Fellow, IEEE) received the PhD degree in computer science from Carleton University, Ottawa, Canada. He is currently a professor and Abdul Kalam Technology Innovation National fellow with the Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. He has authored or coauthored more than 350 scholarly research papers and 12 books. His current research interests include wireless sensor networks and Internet of Things. He has won 11 research paper awards in different Journals and Conferences. He was the recipient of the IEEE ComSoc Asia Pacific Outstanding Young Researcher Award at IEEE GLOBECOM 2012, California, USA, several academic awards and fellowships, such as the Faculty Excellence Award (IIT Kharagpur), Young Scientist Award (National Academy of Sciences, India), Young Systems Scientist Award (Systems Society of India), Young Engineers Award (Institution of Engineers, India), (Canadian) Governor General's Academic Gold Medal at Carleton University, University Outstanding Graduate Student Award in the Doctoral level at Carleton University and the National Academy of Sciences, India – Swarna Jayanti Puraskar (Golden Jubilee Award), Samsung Innovation Awards-2014 at IIT Kharagpur, IETE-Biman Behari Sen Memorial Award-2014, and Careers360 Outstanding Faculty Award in Computer Science for the year 2018 from the Honourable Minister for Human Resource Development (MHRD) of India. Thrice consecutively, he was the recipient of the IEEE Systems Journal Best Paper Award in 2018, 2019, and 2020. He was awarded the Canadian Government's prestigious NSERC Post Doctoral Fellowship and Alexander von Humboldt Research Fellowship in Germany. His team received the GYTI Award 2018 in the hands of the President of India for socially relevant innovations.

Chandranath Chatterjee received the PhD degree from the Indian Institute of Technology Kharagpur, India. He is a currently a professor with the Department of Agricultural and Food Engineering Indian Institute of Technology Kharagpur. His research interests include surface water hydrological modeling, flood inundation modeling, remote sensing and GIS applications in hydrology, and flood forecasting using soft computing techniques. He supervised the theses of several PhD and MTech scholars and has more than 100 international journal and conference publications. He was the past recipient of the Alexander-von-Humboldt Foundation Research Fellowship, Germany, 2005-06. He has received research funding support from several Indian Government funding sources. He is a life member of International Association of Hydrological Sciences (IAHS), Wallingford, U.K.

Shiwen Mao (Fellow, IEEE) received the PhD degree in electrical engineering from Polytechnic University, Brooklyn, NY, USA, in 2004. He is currently a professor and the Earle C. Williams Eminent scholar chair of electrical and computer engineering with Auburn University, Auburn, AL, USA. His research interests include wireless networks, multimedia communications, and smart grid. He is on the Editorial Board of *IEEE/CIC China Communications*, *IEEE Transactions on Wireless Communications*, *IEEE Internet of Things Journal*, *IEEE Open Journal of the Communications Society*, ACM GetMobile, *IEEE Transactions on Cognitive Communications and Networking*, *IEEE Transactions on Network Science and Engineering*, *IEEE Transactions on Mobile Computing*, *IEEE Multimedia*, *IEEE Network*, and *IEEE Networking Letters*. He was a co-recipient of the 2021 IEEE Communications Society Outstanding Paper Award, IEEE Vehicular Technology Society 2020 Jack Neubauer Memorial Award, IEEE ComSoc MMTC 2018 Best Journal Paper Award, 2017 Best Conference Paper Award, Best Demo Award from IEEE SECON 2017, Best Paper Awards from IEEE GLOBECOM 2019, 2016, and 2015, IEEE WCNC 2015, and IEEE ICC 2013, and 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of communications systems. He is a member of the ACM.