Predictive Path Planning of Multiple UAVs for Effective Network Hotspot Coverage

JeiHee Cho, SooMin Ki, and HyungJune Lee, *Member, IEEE*

***Abstract*—In event or disaster scenarios where network communication is jammed, it is important to provide stable network service to users within a reasonable amount of time. We propose a path planning algorithm for unmanned aerial vehicles (UAVs) to serve network traffic in hotspot areas using spatio-temporal information about requests among the region of interest (RoI). The main task of a UAV is to provide communication services to users, while preparing for future hotspots. We propose a simple yet efficient trajectory design consisting of two phases: 1) targeting traffic for a single UAV, and 2) cooperative targeting for multiple UAVs. First, each UAV selects a long-term target considering future traffic and then a short-term target considering the present traffic. When UAVs encounter other UAVs, a cooperative targeting phase ensures UAVs serve traffic in different locations or with different statuses. Our trajectory design enables a UAV to construct its own path for a continuous UAV-enabled network. Simulation and real-world dataset-based experiments confirmed that our targeting scheme provides sufficient network service in a reasonable time, with an average service rate factor of up to 0.85, and an average service completion time relative to the deadline of up to 0.23. The experimental results have demonstrated that our proposed algorithm provides more stable performance compared to other existing algorithms.**

***Index Terms*—Aerial base stations, unmanned aerial vehicle (UAV), network hotspot coverage, path planning.**

I. Introduction

As NUMEROUS information and communication are available almost everywhere, the network connection has become integrated to our daily life, to provide both leisure and essential services. People are becoming more and more familiar with Internet-based services, and are further in need of stable Internet connections. When network connectivity is lost in a disaster or out-of-service due to unexpected network congestion, it must be restored. To tackle this problem, considerable research has been conducted to ensure reliable access and to improve network resilience using subsidiary networks [1], [2], [3] For example, *Project Loon* aims to provide a network connection to the unconnected people using balloons, and launched an Internet-via-balloon service in 2020 [4].

The aerial base station (for example, as in Project Loon) is considered one of the very promising approaches to the provision of connectivity due to its less-constrained deployment. To enable aerial networks to have high flexibility and easy control, Unmanned Aerial Vehicles (UAVs) have been widely applied [5], [6], [7], [8]. UAVs can easily achieve line-of-sight (LoS) links, which allow a large number of ground users to communicate reliably.

Previous works such as [9], [10] solved a similar problem. [9] tries to maximize the minimum throughput in the scenario when a UAV flies in a cyclic path along the cell edge and offloads data traffic for cell-edge mobile terminals. However, the fixed trajectory of UAVs only with varying radius is not sufficient to handle more dynamic spatial variation in network.

In [10], both static users and mobile users are served by a small number of UAVs, considering the fairness among users. As the scheme works in a scenario where fewer UAVs are utilized, the approach is not scalable and is relatively easy to solve. Moreover, similarly to [9], the work does not consider the dynamics of network traffic requests from users, while focusing on providing fair network services.

In case the traffic requests can be estimated to some degree by predicting traffic patterns, there could be a better dynamic traversing approach for mobile UAVs. Traditionally, only time-series prediction has been used for network traffic forecasting, but recent work has used spatio-temporal analysis of data such as cellular traffic [11] or call data records [12] to achieve accurate prediction [13], [14], [15], [16] (for example, as in Fig. 1).

With predictive network traffic methods, UAVs can prepare for future traffic and handle requests within designated deadlines. Accurate predictions of the number of network traffic requests over a region of interest (RoI), enable networks to prepare for hotspot areas.

The main challenge of this work arise at the point that multiple UAVs, still insufficient for full coverage of RoI, need to deal with future traffic requests along with current requests. To effectively mitigate the network congestion in a hotspot area, a UAV needs to arrive at that area beforehand. Given the limited capacity of each UAV to handle traffic, it is essential to develop a resilient trajectory strategy for its future movements, based on projected traffic demands in the near future.

Moreover, as multiple UAVs are deployed for the same mission, it is challenging to discover their optimal trajectories. The

Manuscript received 29 September 2022; revised 29 January 2023 and 6 May 2023; accepted 23 July 2023. Date of publication 27 July 2023; date of current version 19 December 2023. This work was supported in part by the National Research Foundation of Korea funded by the Korea Government (MSIT) under Grant NRF-2021R1A2B5B01002906, and in part by the Institute of Information and Communications Technology Planning and Evaluation funded by the Korea Government (MSIT) under Grant RS-2022-00155966, Artificial Intelligence Convergence Innovation Human Resources Development Ewha Womans University. The review of this article was coordinated by Dr. Chuan Hu. *(Corresponding author: HyungJune Lee.)*

The authors are with the Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea (e-mail: jeihee@ewhain.net; soominki527@ewhain.net; hyungjune.lee@ewha.ac.kr).

Digital Object Identifier 10.1109/TVT.2023.3299302

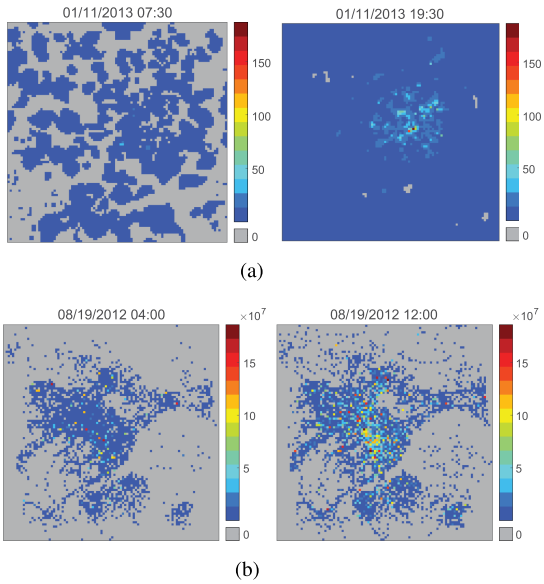

Fig. 1. Example showing spatio-temporal characteristic of call data records (CDR) and cellular traffic in major cities. (a) Call data records distribution in Milano. (b) Cellular traffic distribution in city of China.

UAVs need to make local decisions on the UAV side for their future movement. In case a central server calculates global trajectories for them, it may be more globally feasible but would be too risky upon communication failures. Designing a distributed trajectory planning for each UAV is a key element.

In this article, we present a spatio-temporal network traffic path planning approach for multiple UAVs equipped with base stations, to provide communication to users, while alleviating network congestion in hotspot areas. We solve the problem of handling network traffic demand over the RoI by targeting potential hotspot areas while providing network communication between users in other areas.

We introduce a forward-looking trajectory design of multiple UAVs for UAV-aided wireless networks, providing reasonable service time assuming the deadline of network traffic requests. We leverage future incoming network traffic and remaining network traffic over the RoI to design a path planning algorithm for UAVs that target the hotspot area and traffic demands in a distributed manner.

We propose a targeting scheme for UAVs, which consists of two phases: global targeting and local targeting. We suggest a simple yet efficient targeting scheme that considers both long-term and short-term features. Moreover, we present a cooperative targeting scheme for groups of UAVs to separately serve network traffic in different areas.

The main contributions of this work can be summarized as:

- We present a forward-looking path planning scheme in which multiple UAVs target the essential points of future network traffic, to ease network traffic congestion without manually identifying targets.
- Our approach offers a way to maintain a continuous UAV-enabled wireless network when there are no ground base stations, taking into account service completion time and amount.
- We propose a simple yet efficient scheme that considers both the present and future status of network traffic, to provide effective communication to users, and outperforms other counterpart algorithms.

II. Related Work

UAV networks have widely been deployed to construct a self-organizing network to serve network traffic requests or to act as subsidiary on-the-fly base stations to reduce network traffic congestion. Related work related to UAV-enabled networks can be categorized into UAV deployment and UAV path planning.

A. UAV Deployment

Traditionally, researchers focused on UAV deployment to alleviate the congestion of on-ground base stations at the hotspot event [7], [8], [17]. In [8], the authors investigated the optimal horizontal location of UAVs with some fixed altitude and fixed transmission power to serve the uncovered ground terminals. Also in [17], the researchers find out the optimal altitude of a UAV to maximize the throughput of the ground users within the hotspot area.

While they conducted interesting research about either the horizontal or vertical location of UAVs, there are several approaches that considered the 3D location of UAVs [18], [19]. [18] aims to find the optimal 3D location of UAVs to maximize user coverage. It first calculates the horizontal location of UAVs using an edge prior placement algorithm and then calculates the optimal height of UAVs. Also in [19], they optimized the 3D location of UAVs coupled with prediction of user's location and traffic demand. However, leveraging UAVs as fixed access points does not fully utilize the UAVs irrespective of their flexible maneuverability. Also, static UAV deployment suffers from the lack of fairness and coverage limitation due to the area constraint.

B. UAV Path Planning

To tackle the problem, researchers utilize the less-constrained mobility and high flexibility of mobile UAVs. Some works optimized the 2D trajectory of UAV to offload the data traffic request in three adjacent cell [20] or aimed to maximize the sum rate of UAV-served users with NOMA network [21]. Some recent approaches optimized the 2D trajectory of UAVs [22], [23], [24], [25], [26]. [22] optimized multiple aspects including trajectory and transmit power of UAVs to maximize the uplink data rate of ground users, while [23] suggested a Monte Carlo tree search-based path planning scheme for a single UAV to serve traffic demands from multiple ground mobile users. Moreover, in [24], the authors proposed a trajectory algorithm of a UAV in disaster scenarios with limited resources such as energy and bandwidth and compared their trajectory algorithm with naive approaches like circular, cross, scan, and random.

Beyond the 2D trajectory optimization, 3D trajectory-based UAV path planning schemes are proposed [9], [10], [27], [28]. In [29], the authors investigated a continuous 3D deployment of multiple heterogeneous UAVs, while maximizing the coverage

of the area of interest focusing on the heterogeneity of UAVs. However, the work suggested only the short trajectory of a few UAVs. Also in [30], it aims to design an optimal trajectory of a single UAV and throughput based on reinforcement learning with Internet of Things (IoT) data collection. In [31], the authors applied artificial intelligence to plan energy-efficient 3D paths for three UAVs. Considering the 3D trajectory of multiple UAVs is computationally intensive and makes the problem more complicated. In [28], solved convex optimization problem to achieve global energy efficient trajectory for multiple UAVs for communication with static target ground terminals.

C. Contribution of This Work

The aforementioned previous works considered only the fixed location of the network traffic and aimed to discover the optimal trajectory or deployment of UAVs with some constraints, without considering the realistic characteristics of traffic requests. Furthermore, prior research employed a limited number of UAVs, in a centralized system for computation and management. This approach presents challenges when confronted with unforeseen circumstances. Our work, on the other hand, aims to solve the problem by examining the spatial and temporal characteristics of network traffic requests which are scattered over both local and global regions. Our objective is to propose an efficient 2D trajectory planning method for multiple UAVs in a decentralized manner. Our work is differentiated from existing methods and is capable of accommodating a larger number of UAVs.

More importantly, we consider a practical aspect of time-varying network traffic requests. Since the amount of network traffic increases or decreases as over time, UAVs need to serve the traffic requests within a given deadline, in a timely and spatially effective manner. Our problem scenario is held with a considerable difference against the existing approaches in that they focused rather on the aspect of channel characteristics to achieve a certain quality of service for some fixed fewer users without considering on-demand network services.

As it is important to ensure that network requests are served within a specific deadline, we focus on time-sensitive and location-sensitive path planning of UAVs. We introduce a novel lightweight yet effective target selection scheme for UAVs as aerial base stations to serve network traffic requests over various areas including hotspots.

III. System Overview

We address the problem of forming a subsidiary network from the air using UAVs, so that UAVs can be operated as aerial base stations to serve traffic requests from network devices on the ground. In our scenario, mobile users equipped with devices move around the RoI and generate network requests, and the traffic requests vary spatio-temporally. As the users are crowded at some *hotspots*, where sudden network requests occur frequently, the existing network cannot handle the traffic congestion. To reduce the burden and provide stable network connectivity, additional femtocell deployment using UAVs as aerial base stations can be an effective solution.

We assume that the UAVs are connected to the base station and communicate with each other using wireless radio standards such as IEEE 802.11. Also, it is assumed that a rotary-wing UAV such as quadrotor drone is used, due to its flexible maneuverability and easy hovering. Many references such as [8], [32], [33], [34] have used the rotary-wing UAV to work as a UAV base station. For simplicity, we assume that UAVs and ground users communicate with each other through a line of sight link where the channel quality depends dominantly on the distance between a user and a UAV. This work takes the free-space fading model as a basic channel model; we rather focus on designing an effective UAV path planning algorithm to deal with time-varying dynamically fluctuating network traffics. Thus, it is assumed that once a user is connected with a UAV within its serving capacity in the line of sight, the user can obtain a sufficient quality of service.

The traffic requests are estimated at the level of cells, and UAVs fly over a virtual cell-based grid to serve the traffic. The UAVs are equipped with a Global Navigation Satellite System (GNSS) to be aware of their location and identify the optimal service location. It is assumed that UAVs are homogeneous, with sufficient battery and storage resources since existing UAVs such as SolarXOne can be equipped with solar panels, and the battery charging problem is considered to be orthogonal to our main problem. As this work focuses on more algorithmic path planning, some control issues, such as obstacle collision, collision among UAVs, and navigation are out of scope in this work [35], [36], [37]. We also assume traffic heatmaps are predicted at a ground data center, and UAVs receive the traffic estimate of a cell when a UAV arrives at the cell. The calculated traffic heatmaps are delivered to UAVs periodically or upon request from UAVs.

Our goal is to find an optimal path for each UAV to cooperatively serve location-sensitive traffic requests as much as possible within a designated deadline. We aim to maximize the rate of successfully serviced users and traffic requests, considering the total number of requests and their deadline for both remaining and upcoming traffic. Given predictive network traffic requests [13], [14], [16], [38], we take a mixture of long-term and short-term path planning. When multiple UAVs are encountered within the communication range, they collaboratively decide which targets to visit.

A. Problem Definition

By assuming that the prediction for future traffic request is given to UAVs, our goal is to maximize the amount of served traffic request on time. We want to jointly optimize the trajectory of UAVs and on-demand traffic request. The traffic request occurred in timeslot t is denoted by $tx_{cell}(t) = [tx_{cell}, D_t]$, where tx_{cell} is updated as the UAV serves the traffic request, and D_t denotes the remaining deadline of the traffic request. UAVs should serve traffic request before D_t reaches at zero. As we assume UAVs can only hover over the virtual grid points, UAVs need to select a cell of which traffic requests to serve first, with regard to the deadline or their future trajectory. We also assume that UAVs can serve traffic request with a fixed capacity, due to the bandwidth limitation. Let us denote the location of UAV at

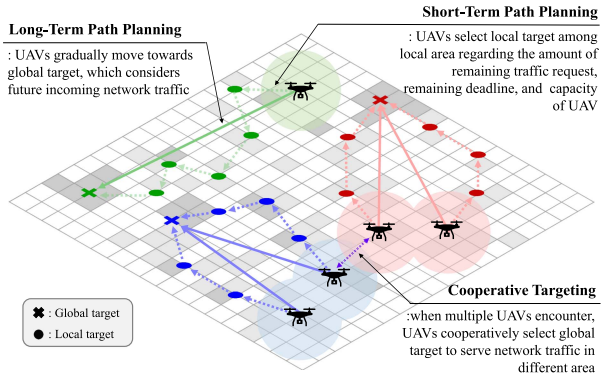

Fig. 2. Our UAV-aided network coverage scheme where the same color circled drones are in the same group.

timeslot t as $G_u(t) = [x_u(t), y_u(t)]$, and cells that UAV served at $G_u(t)$ as $servedCell(G_u(t))$. Every UAV tries to maximize the amount of total served traffic during the active time, with regard to the capacity of a UAV (CAP). The objective function 1 is to maximize the total served traffic within the total active time of a UAV, while the UAV calculates its own trajectory in a distributed manner, without any help from a central control station.

$$\max \Sigma_{t=1}^{T} \Sigma_{u \in UAV} \Sigma_{cell \in servedCell(G_u(t))} tx_{cell}(t),$$
$$\text{subject to } D_t \geq 0,$$
$$\Sigma_{cell \in servedCell(G_u(t))} tx_{cell}(t) \leq CAP. \quad (1)$$

This objective is hard to achieve due to the following two main reasons. First, we do not use a control station to calculate and manipulate the UAVs. Instead, UAVs need to plan their own trajectory with their own information with a given limited information. Second, even the future network traffic heatmap is provided, it is hard to check all the possible locations to go and select the optimal trajectory since there are too many options to consider. To achieve a feasible solution based on the above formal problem definition, we aim to solve this problem with a heuristic algorithm-based approach.

B. Procedure

Our UAV network to assist network traffic consists of two phases: 1) the design of the trajectory of each UAV, and 2) cooperative targeting of encountered UAVs. A high-level overview is shown in Fig. 2.

1) UAV Trajectory Design: To effectively alleviate network congestion, it is imperative to ensure that UAVs are equipped with the necessary capabilities to handle sudden traffic requests. As such, it is recommended that UAVs exhibit foresight while keeping track of current requests. We develop a hybrid path planning for UAVs to select a hotspot within a larger range as the long-term target, namely the global target, while visiting the short-term target, namely the local target, on the way to the global target. More details are provided in Section IV.

2) Cooperative Targeting: When multiple UAVs encounter each other, they jointly select the targets in their area to ensure higher throughput by reducing duplicate coverage. Given that traffic requests can be prioritized based on their remaining deadlines, it is feasible for a maximum of two nearby UAVs to serve requests with different priorities. If the number of UAVs exceeds two, UAVs can take part in the different areas by selecting different global targets. A detailed cooperative targeting procedure is described in Section V. The overall procedure is described in Algorithm 1.

Algorithm 1: Overall Procedure of UAV Networks.

```
// I. Serve network traffic based on priority
serveTraffic();
// II. Transition to long-term and short-term targeting for
cooperative targeting upon encounter of new UAVs
if isGrouped() == True then
  // If this UAV is grouped, do cooperative targeting
  if Encountered UAVs in communication range then
    if Encountered UAV is a group UAV then
      Invoke longTerm() to cooperatively select a global
      target;
    else
      Invoke longTerm() to select a new global target
      for the solo UAV;
    end if
    Invoke shorTerm() for each UAV to select a local
    target;
  else
    if selectGlobal() == True then
      Change the mode of UAV into the solo UAV mode;
      Invoke longTerm() for each UAV to select a new
      global target;
    else
      Invoke shortTerm() for each UAV to select a local
      target;
    end if
  end if
else
  // If this UAV is not grouped, do cooperative targeting
  only if other neighbors exist
  if Encountered UAVs in communication range then
    if Encountered UAV is a group UAV then
      Invoke longTerm() to select a new global target;
    else
      Form a group with one of the encountered UAVs;
      Set the mode of UAVs to the group mode, and
      assign one UAV to serve nearby traffic with the
      shortest deadline;
      Invoke longTerm() to select a new global target for
      the group UAV;
    end if
    Invoke shortTerm() for each UAV to select a local
    target;
  else
    if selectGlobal() == True then
      Invoke longTerm() for each UAV to select a new
      global target;
    else
      Invoke shortTerm() for each UAV to select a local
      target;
    end if
  end if
end if
```

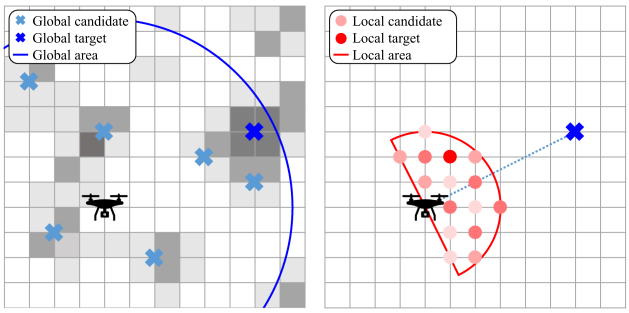

Fig. 3. Examples of selecting global and local targets, in which the gray level of a cell in panel a) implies the value of the cell, and the color of the circle in panel b) indicates the score of the grid point. (a) Long-term selection. (b) Short-term selection.

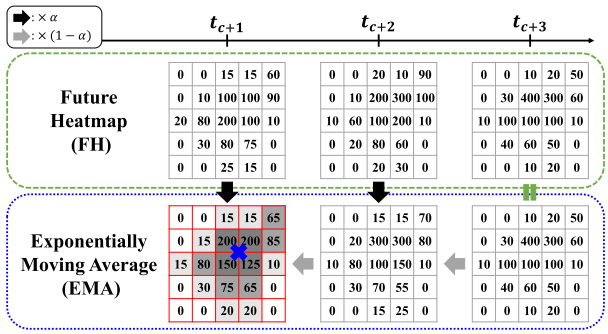

Fig. 4. Example of a weighted heatmap calculation with the *window* size of 3 and $\alpha = 0.5$. The finally selected grid point is denoted as marker X.

IV. UAV Trajectory Design

When there are fluctuations in network traffic demand due to temporary events such as sport events or festivals, it is important to provide stable and scalable communication channels to ease network congestion. The traffic requests from each user should be served in a reasonable time, and thus, it is important for UAVs to be in the hotspot area on time, while serving network traffic near the path along the hotspot area.

We attempt to leverage future traffic hotspots together with the remaining network traffic and its deadline. Deciding the trajectory of UAVs with incomplete or short-sighted information may result in the failure to detect future hotspot events, as the UAVs may not be capable of fully surveying the entire RoI. Therefore, it is important for UAVs to first generate their own path by selecting a long-term goal based on predicted future estimates, and then selecting short-term goals using the current status along with the long-term goal. A conceptual overview of the path planning is depicted in Fig. 3.

A. Long-Term Path Planning

To generate the trajectory of the UAVs that handle network congestion, the UAVs need to select a location to visit based on future traffic estimates. For long-term path planning, each UAV selects its own global target using a given number of future heatmaps. Our long-term path planning consists of two phases: 1) weighted network traffic heatmap generation that allows the prediction of future traffic information, and 2) global target selection.

1) Weighted Network Traffic Heatmap Generation: Since it is time consuming to check all future traffic heatmaps every time, it would be needed to identify some critical grid points and check out the amount of network traffic over a certain time period in the near future. As mentioned, UAVs periodically obtain the future traffic heatmap estimates or can also retrieve them upon request. Based on the given predicted future traffic heatmaps, we merge them into a *weighted traffic heatmap* using an exponential moving average (EMA). In general, the EMA is a useful indicator that gather information of recent data points and past ones, while placing a more weight on the recent ones, and predict the value of next step. To apply this into the given setting of current and future data points, we modified the EMA in a backward manner so that both current data traffic and the future traffic estimates can be integrated as the expected traffic requests. With a weighted heatmap calculated by the EMA, we obtain a summarized information of future traffic requests.

Some more details about how the weighted heatmap is calculated are described in the first part of Algorithm 2. The computational complexity for calculating a weighted heatmap is given by $O(window)$ for accumulate a series of heatmaps from consecutive $window$ timeslots.

Fig. 4 shows an example of calculating a *weighted heatmap* with the *window* size of 3, and the current timeslot of c. We consider the farthest future heatmap within the *window*, called the future heatmap (FH) at t_{c+3}, which is considered to be the initial EMA map. The next EMA map at t_{c+2} is calculated using the EMA map at t_{c+3} and the FH at t_{c+2}. The equation to derive the EMA map is as follows:

$$EMA_{c+1} = \alpha \cdot FH_{c+1} + (1-\alpha) \cdot EMA_{c+2}, \qquad (2)$$

where FH_{c+1} is the future heatmap at t_{c+1}, EMA_{c+1} is the computed EMA value until t_{c+1}, and α is a smoothing factor. As the standard EMA usually sets α to be $2/(N+1)$ where N is the size of sliding window, we also uses $2/(window+1)$ as α. A UAV at t_c uses EMA_{c+1} as its *weighted heatmap*.

2) Global Target Selection: After calculating the aforementioned weighted traffic heatmap, each UAV selects its own global target in the global area. The global area consists of grid points that are available to visit over the period of *window*, as visualized in Fig. 3(a), in which the darkest area is selected as the global target. We predict the occurrence of network traffic hotspots using the weighted traffic heatmap, by calculating the score of each grid point in the RoI. The score of a grid point is the summed value of cells within the communication range, where the value of cells is the score in the weighted traffic heatmap of the same cell. For example, in Fig. 4, the grid point with marker X is selected, since the summed value is the highest when UAV can cover four cells from the center. Normally, when the period of *window* is expired, the global target is refreshed and newly selected. However, UAV can keep its global target when it has the remaining traffic around the current global target. By employing the adaptive global target selection, UAVs can deal

Algorithm 2: LongTerm() for a Solo UAV.

Input: t_c: current timeslot, $window$: future window period,
$heat_t$: traffic heatmap at time t, α: smoothing factor
Output: $globalTarget$
// I. Calculate a weighted heatmap
Initialize WH;
$WH_{t_c+window} = heat_{t_c+window}$;
for ts in range (t_c + $window$-1, t_c + 1) **do**
  $WH_{ts} = (1-\alpha) \cdot WH_{ts+1} + \alpha \cdot heat_{ts}$;
end for
// II. Select a global target in the global area
Initialize $score$;
for grid point $gp \in visitableGridList(window)$ **do**
  for cell $cell \in coverableCellList(gp)$ **do**
    $score(gp) = score(gp) + WH_{t_{c+1}}(cell)$;
  end for
end for
if there are multiple grid points with the maximum $score$ **then**
  Randomly select one among them as $globalTarget$;
else
  $globalTarget =$ a grid point with the maximum $score$;
end if

Algorithm 3: ShortTerm() for a Solo UAV.

Input: t_c: current time, d: deadline of traffic,
$heat_t$: traffic heatmap at time t, GT: global target, U: this UAV,
loc_x: location of grid point or UAV x, cap: capacity of this UAV,
$tx_{cell,t}$: traffic request in cell $cell$ at time t,
w_t: weight for traffic demand at time t
Output: $localTarget$
// I. Set the local area of UAV
Initialize $localArea$;
for grid point $gp \in visitableGridList(1)$ **do**
  if $\overrightarrow{loc_U loc_{GT}} \cdot \overrightarrow{loc_U loc_{gp}} \geq 0$ **then**
    Include gp to $localArea$;
  end if
end for
// II. Select a local target in the selected local area
Initialize $score$;
for grid point $gp \in localArea$ **do**
  $remCap = cap$;
  for ts in range (t_c - d + 1, t_c + 1) **do**
    for cell $c \in coverableCellList(gp)$ **do**
      $score(gp) = score(gp) + min(tx_{cell,ts}, remCap) \times w_t$;
      $remCaps = max(0, remCap - tx_{cell,ts})$;
    end for
  end for
end for
if there are multiple grid points with the maximum $score$ **then**
  Randomly select among them and set as $localTarget$;
else
  $localTarget =$ a grid point with maximum $score$;
end if

with heavy network traffic demand that cannot be served in one shot otherwise, by continuously visiting some immediate areas.

The overall procedure for long-term path planning is described in Algorithm 2. The worst-case computational complexity for selecting a global target is $O(|visitableGridList(window)| \times |coverableCellList(gp)|)$ to check all of possible grid points in the global area.

B. Short-Term Path Planning

In long-term path planning, UAVs produce a path by considering possible future traffic heatmaps. Based on the long-term guideline, each UAV selects its own local target with regard to the global target and the remaining network traffic at the timeslot within the radio range of the UAV, for short-term path planning. We investigate an effective way in which to select local targets while continuously moving towards the global target. Our short-term path planning consists of three phases: 1) local area selection, which decides the direction in which UAVs gradually move to the global target, 2) local target selection, and 3) serving traffic at the target location.

1) Local Area Selection: After the global target is selected, UAVs re-scale their own local area with respect to the global target. The local area of a UAV consists of the grid points that the UAV can visit in one timeslot, in the circular area with the current location at the center. To make a UAV move towards the global target, each UAV reshapes its own local area into a half-circle shape, as shown in Fig. 3(b). In Fig. 3(b), the half-circle drawn in the red color is the local area of UAV, and the red and pink colored circles are candidates for the local target.

When the global target is accessible in one timeslot, the local area of the UAV expands a circle, and the UAV serves traffic near the global area until the *window* gets expired or there are no remaining traffics.

2) Local Target Selection: During the short-term path planning phase, UAVs select a grid point to visit for the next timeslot as the local target among the candidates within the local area. Our scheme considers three factors: 1) the capacity of the UAV, 2) the remaining deadline of the network traffic, and 3) the number of network traffic requests. Our local targeting scheme first calculates the score of each grid point based on the amount of network traffic demand that the UAV can serve within the remaining deadline of traffic requests. We design a deadline-weighted scoring system, which gives more weight to network traffic requests with shorter remaining deadlines. First, the weight for each traffic demand occurring at timeslot t is calculated as follows:

$$w_t = 1 - \frac{remainingDeadline}{deadline}, \tag{3}$$

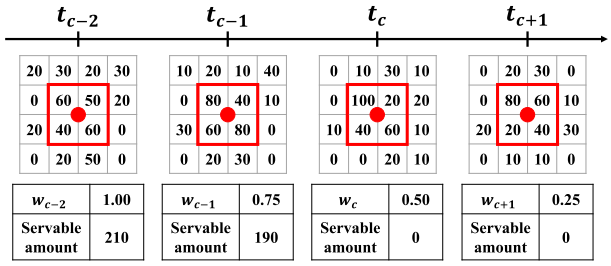

Fig. 5. Example of our scoring system in which a local optimal is selected, when the deadline is set to four, the capacity of the UAV is 400, the number inside the cell indicates the number of traffic requests, and coverage of the UAV is four cells around the selected grid point.

where $remainingDeadline$ is the value of the remaining deadline, and $deadline$ is the initial deadline of the traffic when it first occurred. Since the selected local target will be visited in the next timeslot, the UAV should consider the status of network traffic from the perspective of the next timeslot. Therefore, UAV should consider the heatmap from timeslot $c+1$ to $c-deadline+1$, where c is the current timeslot. Also, $remainingDeadline$ is calculated at the timeslot of $c+1$.

To maximize UAV usage, it is necessary to select the grid point that generates an amount of network traffic matching the UAV's capacity. When a UAV moves to a grid point, it is important to check how much network traffic that UAV can serve within a specific deadline. Our scoring system therefore checks the traffic requests for each heatmap. Starting from the heatmap with the shortest remaining deadline, the UAV checks how much network traffic can be served. If the traffic request from the heatmap is larger than the UAV's capacity, the UAV ignores the excess amount. The UAV scores a grid point by multiplying the weight of the heatmap and the traffic request of the heatmap and sums up the multiplicative quantity until the cumulative traffic request exceeds the capacity of the UAV. The score for a grid point G is calculated as follows:

$$score(G_i) = \Sigma_{i=d}^{0}(w_{c+1-i} \times \min(RC, \Sigma_{cell \in coveredCell(G_i)} tx_{cell})), \quad (4)$$

where d is the $deadline$, tx_{cell} is the traffic request of the cell, and RC is the remaining capacity of the UAV after subtracting the sum of tx_{cell}. After scoring all grid points in the local area, the UAV selects the one with the highest score. If there are multiple grid points with the highest score, the UAV randomly selects one grid point from them. Fig. 5 shows an example of the calculation of the score for a grid point when the $deadline$ is 4, the capacity of the UAV is 400, and the UAV coverage is 4 cells from the center. The weight for the heatmap in $(c-2)ts$ is 1.00, since its $remainingDeadline$ is 0 in $(c+1)$ ts. The weights for the other heatmaps are 0.75, 0.50, and 0.25, as in Fig. 5. Using the weight values obtained, the final score for this grid point is $1.00 * 210 + 0.75 * 190 + 0.5 * 0 + 0.25 * 0 = 352.5$. The network traffic demand from h_c and h_{c+1} is not considered in this case, since the network traffic requests from the past two heatmaps already exceed the UAV's capacity. Considering both the deadline of the network traffic and the UAV's capacity, the UAV selects an effective grid point to visit for the next timeslot in a simple, greedy manner.

The detailed short-term procedure is described in Algorithm 3, and a conceptual image is provided in Fig. 3(b). The computational complexity to choose a local area is $O(|visitableGridList(1))$, and $O(|localArea| \times deadline \times |coverableCellList(gp)|)$ in the worst-case to find out a local target to visit.

C. Serving Traffic

When a UAV moves to a selected location, it needs to select the order of cells to serve. The cells within all four corners inside the UAV's radio range are candidates. The priority of the cells is organized based on their direction to the global target. Since the UAV gradually moves toward the global target, cells along the way towards the global target have a chance of being revisited by the UAV. The cells opposite to the global target have a lower or no chance of the UAV's revisit. Therefore, the UAV first serves network traffic that has the shortest deadline, and after, selects the cells that are located opposite to the global target. The priority of network traffic service is set as 1) the cell with the shortest deadline, and 2) if the deadline is same, the one which is the farthest from the global target is selected.

V. Cooperative Service Targeting

While the UAV moves around the RoI based on a planned path, if the UAV discovers one or more UAVs in its vicinity, it shares information with its neighbor, and begins a cooperative targeting scheme. In the cooperative targeting procedure, it forms a group with the neighboring UAV, which have first responded to the packet. However, if a UAV is currently a member of a group, and meets another UAV, they begin negotiation to separate the traffic service area. When multiple UAVs encounter, a UAV with the smallest device ID decides to be the head of a UAV groups with the encountered neighboring UAVs. The group UAV consists of at most two UAVs. The UAV group remains as a group until the allocated time to the global target remains. Once the allocated time is over, the group is finally disassembled.

In this section, we present a scheme for cooperative traffic targeting by multiple UAVs which involves two categories: 1) forming a group of UAVs, and 2) negotiation between multiple UAVs. The overall procedure of cooperative targeting is described in Algorithm 1, while the long-term and the short-term procedure of cooperative targeting is provided in Algorithms 4 and 5, respectively.

A. Cooperative Operation of UAVs

In order to fairly cover network traffic throughout the RoI, UAVs would rather serve network traffic separately for each different area. To deal with the property of local traffic demand dynamics, we let at least two UAVs operate together within a group for efficient coverage. We limit two UAVs to form one group as there are two ways to prioritize the network traffic request: 1) the one with more remaining deadline and 2) the one

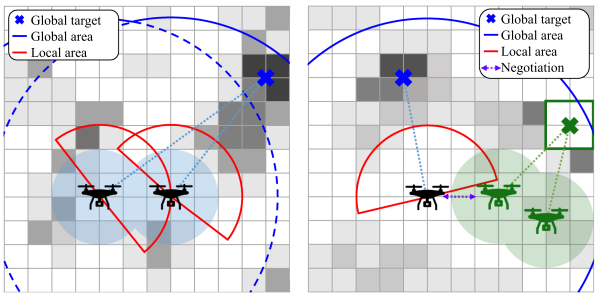

Fig. 6. Example of cooperative target selection in which UAVs indicated by the same color form a group. (a) Cooperative targeting of a UAV group. (b) Negotiation when a grouped UAV encounters a single UAV.

Algorithm 4: Cooperative LongTerm() for Multiple UAVs.

Input: $this$: this UAV, $neigh_{this}$: neighboring UAV of this UAV, loc_x: location for grid point or UAV x, GT_u: global target of UAV u
$longTermExcept(GlobalTargetLocation)$: Zero out the current global target location and return another global target
// I. If current UAV is grouped
if *isGrouped(this)* $==$ true **then**
  for UAV $u \in neigh_{this}$ **do**
    if *isGrouped(u)* $==$ true **then**
      Invoke *negotiation()* to select a global target;
    else
      Invoke *longTermExcept*(GT_{this}) to select a new global target for u;
    end if
  end for
  // II. If current UAV is solo, form new group or target new global target
else
  for UAV $u \in neigh_{this}$ **do**
    if *isGrouped(u)* $==$ true **then**
      Invoke *longTermExcept*(GT_u) to select a new global target for $this$;
    else
      Form a group with UAV u;
    end if
  end for
end if

with less remaining deadline. Additionally, when multiple UAVs form a group, they must take turns selecting the next location to prevent overlapping between the UAVs. However, when only two UAVs are involved, they are able to perform calculations simultaneously using different criteria.

The UAV group formation procedure starts upon an encounter between two UAVs. When two UAVs form a group, they cooperatively select a global target over the global area as illustrated in Fig. 6(a). One UAV with smaller ID selected to compute the list of global target candidates for the group. The global area of the UAV group is a union of the global area of each UAV. One

Algorithm 5: Cooperative ShortTerm() for a Group UAV.

Input: t_c: current time, d: deadline of traffic, $heat_t$: traffic heatmap at time t, GT: global target, U: this UAV, loc_x: location of grid point or UAV x, cap: capacity of this UAV, $tx_{cell,t}$: traffic request in cell $cell$ at time t, w_t: weight for traffic demand at time t
Output: $localTarget$
// I. Set the local area of UAV
Initialize $localArea$;
for grid point $gp \in visitableGridList(1)$ **do**
  if $\overrightarrow{loc_U loc_{GT}} \cdot \overrightarrow{loc_U loc_{gp}} \geq 0$ **then**
    Include gp to $localArea$;
  end if
end for
// II. Select a local target in the selected local area
Initialize $score$;
if U $==$ ReverseUAV **then**
  // If this UAV is *ReverseUAV*, use *scoreRev*
  for grid point $gp \in localArea$ **do**
    $remCap = cap$;
    for ts in range (t_c + 1, t_c - d + 1) **do**
      for cell $c \in coverableCellList(gp)$ **do**
        $score(gp) = score(gp) + min(tx_{cell,ts}, remCap) \times w_t$;
        $remCaps = max(0, remCap - tx_{cell,ts})$;
      end for
    end for
  end for
else
  // If this UAV is not *ReverseUAV*, use *score*
  for grid point $gp \in localArea$ **do**
    $remCap = cap$;
    for ts in range (t_c - d + 1, t_c + 1) **do**
      for cell $c \in coverableCellList(gp)$ **do**
        $score(gp) = score(gp) + min(tx_{cell,ts}, remCap) \times w_t$;
        $remCaps = max(0, remCap - tx_{cell,ts})$;
      end for
    end for
  end for
end if
if there are multiple grid points with the maximum $score$ **then**
  Randomly select among them and set as $localTarget$;
else
  $localTarget =$ a grid point with maximum $score$;
end if

UAV calculates a score of each grid point over the global area and selects a global target for the group for the incoming time period. The grouped UAVs move towards to the same global target, while individually selecting and serving their own local target. To enhance efficient collaboration, the network traffic priority is set differently between the group members. To balance the coverage priority between the short-term and the long-term

traffic requests, we assign one UAV randomly to prioritize the oldest requests first, while another UAV is assigned for serving the latest requests first.

To make a UAV intentionally short-sighted to serve the traffic requests with the priority of shortest deadlines, the UAV called *ReverseUAV* adopts a reverse weighting in the deadline-weighted scoring system as in (5). The score of grid point G_i in the local area of *ReverseUAV* is calculated as follows:

$$scoreRev(G_i) = \Sigma_{i=0}^{d}(1 - w_{c+1-i} \times \min(RC, \Sigma_{cell \in coveredCell(G_i)} tx_{cell})), \quad (5)$$

where d is the traffic deadline, tx_{cell} is the number of traffic requests from a cell, and *RC* is the remaining capacity of the UAV after subtracting the sum of tx_{cell}. For example, the weight for each heatmap in Fig. 5 is 0.25, 0.5, 0.75, and 1.00, respectively, for the heatmaps in $c-2$, $c-1$, c, and $c+1$. The score of the grid point is 1.00 * 200 + 0.75 * 200 + 0.5 * 0 + 0.2 * 0 = 250. In case that two UAVs use the same scoring system, they would likely choose the same local target. To lessen the chance of selecting the same local target, we use this reverse scoring system for UAVs to cover an area more fairly under various traffic serving deadlines.

B. Negotiation of Multiple UAVs

When a UAV group navigates over the RoI, there are many chances to meet other UAVs. It is inefficient for all of the UAVs to form a single group whenever they encounter new UAVs. This is because multiple UAVs will serve only a few network traffic requests, due to redundant visits and the targeting of a single global target. To balance the visiting area, we introduce a negotiation between multiple UAVs to serve discrete areas.

There are two cases in which multiple UAVs might encounter each other: 1) when a group UAV encounters multiple group UAVs; and 2) when a group UAV encounters a solo UAV. First, each group UAV gets to select one global target amongst the union of the UAVs' global area. We select the same number of global targets as the number of groups, selecting N global targets when N UAV groups are encountered. When the first global target is selected, we zero out the circular shaped local area around the first global target, and then the next global target is selected. If the whole area is zeroed out, and there are still some groups that try to select a global target, they randomly select global targets in the global area. After selecting N global targets, a global target is assigned based on the distance to each group's center, to reduce the travel time. In the second case, when a group UAV meets solo UAVs, only the solo UAVs reselect a global target amongst its own global area. If the group UAV's global target is inside the solo UAV's global area, the UAV zeros out the circle area around the global target to avoid selecting global targets in the similar area, and then selects its global target. This negotiation between multiple UAVs increases the opportunities to serve network traffic in various areas. A high level description of the negotiation between UAVs is shown in Fig. 6(b), in which the green colored group UAVs encountered a solo UAV.

TABLE I
SIMULATION ENVIRONMENT AND PARAMETERS

Experiment Environment	
Territory Area	$1 \times 1\ km^2$
Grid Size	$10\ m$
Timeslot (ts)	3 sec
UAV Capacity	4000 $byte$
UAV Speed	$12\ m/s$
UAV Altitude	$100\ m$
# of UAVs	15∼35
Communication Radio Range	110∼ m
Simulation Parameter	
Global *Window*	5 ts
Traffic *Deadline* (d)	10 ts

VI. EVALUATION

A. Experiment Environment

We validate our proposed scheme using different datasets over the RoI of 1×1 km^2 with a virtual grid topology of 100×100 cells with the cell size of 10×10 m^2 as shown in Fig. 8. The initial location of the UAVs is set to be the center of the RoI, and the flying speed of a UAV is assumed to be 12 m/s, with an altitude of 100 m. As we focus on the coverage for on-ground users from UAVs in the air, the altitude can be modified by adjusting the transmission power at each UAV.

The capacity of a UAV is defined as the amount of downstream traffic bytes that the UAV can accommodate at each timeslot in maximum, and is set to 4,000 bytes in the experiments. The duration of a timeslot, (ts), is set as 3 sec, and the UAVs can only select the grid point as a next location. The design parameters of $window$ and $deadline$ are set to 5 and 10 timeslots, respectively. More information about parameter values is listed in Table I. We evaluate network traffic service performance in terms of average service rate, amount of on-time serviced traffic, the number of serviced grids, and traffic served time relative to the expected deadline. Our experiments validate our scheme based on simulated dataset and real-world dataset.

B. Simulation Dataset-Based Validation

We validate our local and global target selection, and cooperative targeting scheme using four counterpart algorithms with a simulation dataset: 1) *Ours:* our proposed scheme that follows the algorithms described above, 2) *LocalRandom:* randomly selects a next location among local area, 3) *LocalGreedyW:* a partial version of *Ours* that selects the local target scoring method as in *Ours* without considering the global target, 4) *GlobalD:* a partial version of *Ours* that only selects the global target as in *Ours*, while selecting a local target that is the closest to the global target, and 5) *Ours (w/o neg):* our scheme that considers both local and global target selection, but without applying the cooperative targeting.

The simulation dataset that is used in this experiments came from [14], in which the network traffic traces of 470 access points (APs) are recorded in the University of Oulu, Finland. This data includes the AP IDs, dates of data collection, the number of users, received data bytes, transmitted traffic data, and location

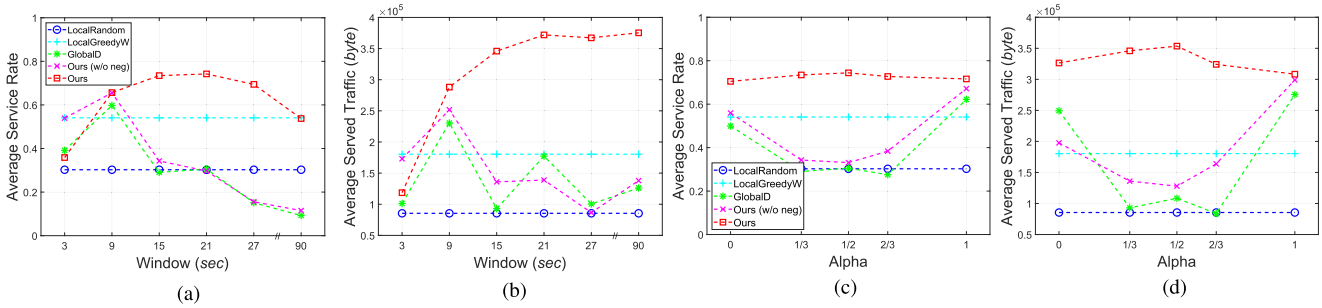

Fig. 7. Performance dynamics with regard to internal parameters of *window* and *alpha* with a *deadline* of 30 seconds. (a) Average service rate with different *window*. (b) Average served traffic with different *window*. (c) Average service rate with different *alpha*. (d) Average served traffic with different *alpha*.

names of each AP. The time interval between two consecutive records is 10 minutes, and the total number of records is 7,586. Since this dataset does not provide the actual location of the APs, we manually produced a dataset by assigning random positions to the APs on top of the original dataset. Considering spatial and temporal characteristics of network traffic, we categorized the type of network traffic as: 1) $Spot$: traffic requests that have occurred in only one cell, without grouping, 2) $Local$: traffic requests generated in an area with a random width and height, across 1 to 5 cells, and 3) $Heavy$: traffic requests across 5×5 cells.

Every type traffic is requested for a random duration in a range of 10 ts, and the number of occurrences of each type per heatmap is in the range of 1 to 3 for $Heavy$, 1 to 4 for $Local$, and 1 to 30 for $Spot$. The traffic value for each cell is selected from [14], selecting compatible traffic value after a fixed start time. We constructed three datasets named *Enterprise-Network1* consisting only of *Heavy*, *Enterprise-Network2* consisting of *Heavy* and *Local*, and *Enterprise-Network3* consisting of *Heavy*, *Local* and *Spot*. The standard deviation of the dataset is 4.72, 11.49, and 11.61 for *Enterprise-Network1*, *Enterprise-network2*, and *Enterprise-Network3*, respectively. Since the original values from the data are the traffic bytes requested from the APs, we rescaled each modified data point to have the maximum value of 6,000. A visualization sample of the simulation dataset in our experiments is shown in Fig. 8.

We first investigated the effects of parameter settings in our scheme using *Enterprise-Network3* data. In Fig. 7, we show performance dynamics as an internal parameter of *window* or α varies. It should be noted that *LocalRandom* and *LocalGreedyW* are not affected by the parameters since they do not consider future values. In Fig. 7(a) and (b), we varied the parameter of *window* from 1 to 9 and 30 timeslots (corresponding to 3 to 27 and 90 seconds), which can cover the whole RoI. We measured the average service rate and the average served traffic. A total of 10 trials were conducted to quantify the average performance. The average service rate is calculated as the average of $(servedtraffic)/(requestedtraffic)$ for every cell that has requested traffic in each trial. The average served traffic is quantified as the average of the sum of served traffic in each trial.

In case of *window* of one timeslot, a selected global target is accessible within one timeslot, and thus, the local area is

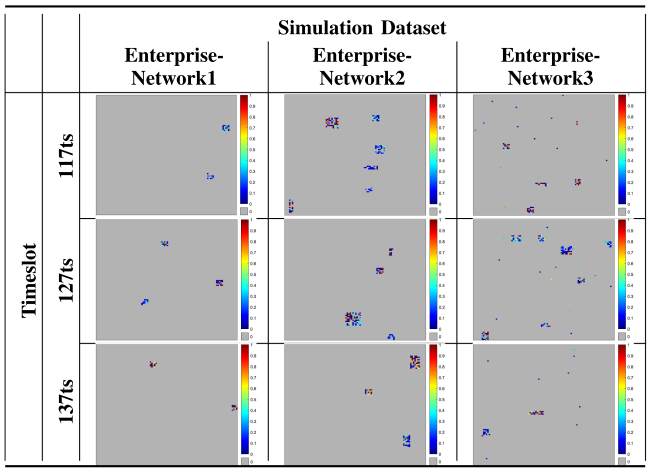

Fig. 8. Visualization of our simulated dataset, *Enterprise-Network1*, *Enterprise-Network2*, and *Enterprise-Network3* at selected timeslots, showing relatively high traffic requests in red and low traffic requests in blue.

set as a circle. Since a global target is selected from a small area, the UAVs cannot get the benefit of anticipating the future values, and thus, the UAVs only move locally, leading to low performance in both average service rate and the number of served data bytes. When *window* is large enough to cover the whole RoI, for example 90 sec in this experiment, the UAVs in a discrete location can select the same global target, leading to a decrease in average service rate performance. Moreover, UAVs can not get chance to negotiate, since even UAVs are located in discrete grid points they have no chance to communicate with others but still select the same global target due to the large global area size. However, the UAVs can still select hotspot areas and serve heavy traffic. When UAVs move closer to a selected global target, they will reselect the global target, which leads to no decrease in the total served traffic performance. *Ours (w/o neg)* and *GlobalD* show a decrease in the performance as *window* increases because these two schemes do not use cooperative targeting and make UAVs select their own separate global target. It implies that selecting proper *window* size is important since it effects the target selection. When *window* is small, both average service rate and average served traffic shows poor since the local and global area is limited, and when *window* is large only average served traffic performance sustains by selecting heavy traffics.

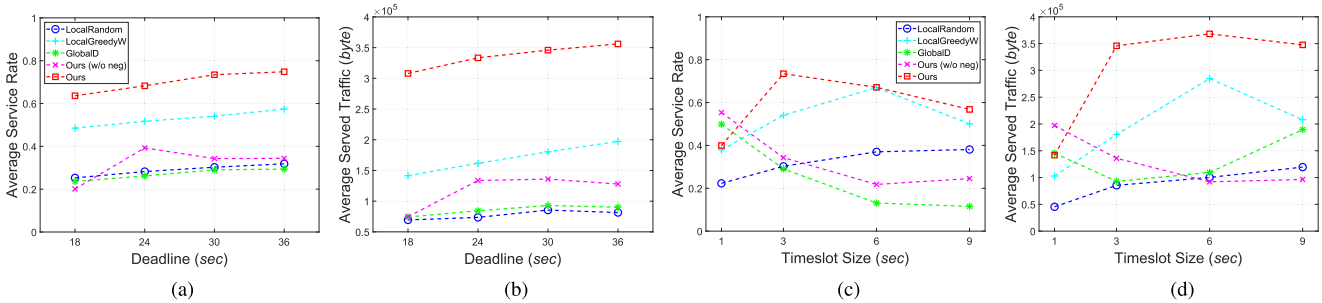

Fig. 9. Performance dynamics with regard to system parameters of *deadline* and *timeslot* size. (a) Average service rate with different *deadline*. (b) Average served traffic with different *deadline*. (c) Average service rate with different *timeslot*. (d) Average served traffic with different *timeslot*.

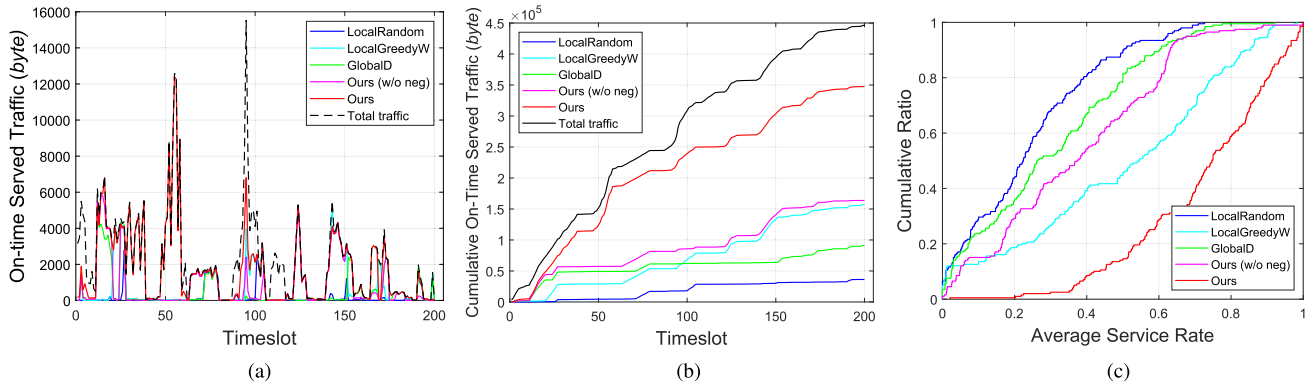

Fig. 10. Performance dynamics of the amount of served traffic and cumulative distribution function of served rate of each timeslot with 30 UAVs. (a) Amount of served traffic per heatmap. (b) Cumulative growth of served traffic. (c) Cumulative ratio of average service rate.

We vary *alpha* from 0 to 1, using *Enterprise-Network3* in Fig. 7(c) and (b). As previously mentioned, α is originally calculated as $2/(window + 1)$ and thus, we mainly use $\alpha = 1/3$. *Ours* shows a steady performance as α varies, while others show some more fluctuation. Using 0 as α indicates that the algorithms consider only the furthest future heatmap, while using 1 as α implies that they can only consider the nearest future heatmap. If When various deadline requests are given, effective negotiation procedures are required to sustain high service rate and served traffic amount if compared *Ours* with *Ours (w/o neg)*. It implies that irrespective of how traffic requests with different deadlines need to be prioritized, our algorithm shows high and stable performance.

Lastly, using a fixed internal parameter setting, i.e., α of $1/3$ and *window* pf 5 timeslots, we varied system parameters of *deadline* and *timeslot* size to validate performance. We varied *deadline* from 6 to 12 timeslots using *Enterprise-Network3* data, and measured the average service rate and average served traffic (Fig. 9(a), (b)). If *deadline* becomes relaxed, UAVs have more chances to serve network traffic, and thus, the overall performance increases for all algorithms, while our scheme significantly outperforms the others. This implies that although the *deadline* parameter affects performance, it is obvious that using longer *deadline* gives more chance for UAVs to serve the traffic requests, enhancing performance. We also modified the timeslot size from 1 to 9 seconds in Fig. 9(c) and (d). When the *timeslot* size is small, UAVs move more passively due to the shrunken local area. When *timeslot* size is large, the size of both global and local area is extended, enhancing the opportunity to select more heavy traffic requests with priority. Similar to the *window* experiment, when *timeslot* size is too large, the global area may cover the whole RoI and have some duplicated global target selection, leading to a decrease of the average service rate performance.

We investigated how the amount of served traffic increases over time using 30 UAVs in *Enterprise-Network3* as in Fig. 10(a). We visualized the total flow of network traffic requests in each timeslot as a dotted line, and the other algorithm's performance as a solid line with each different color. Each solid line shows the amount of network traffic served for the network traffic that has occurred in the timeslot. *Ours* showed an amount of served traffic very similar to *Total traffic*. The traffic requests between 27 ts to 80 ts are almost perfectly served, while others showed relatively poorer performance. In Fig. 10(b), the amount of served traffic is plotted in a cumulative way, showing total served traffic in the final timeslot. *Ours* shows the highest amount of total served traffic, of 347,421 bytes, which is 77.9% of the total traffic, and 183,683 bytes higher than the next best performing algorithm, *Ours (w/o neg)*. By comparing *Ours (w/o neg)* and *GlobalD*, it is clear that the short-term selection in our scheme plays a key role because the amount of served traffic of *Ours (w/o neg)* outperforms that of *GlobalD*. This result indicates that selecting the proper local targets is essential for serving especially heavy traffic demand.

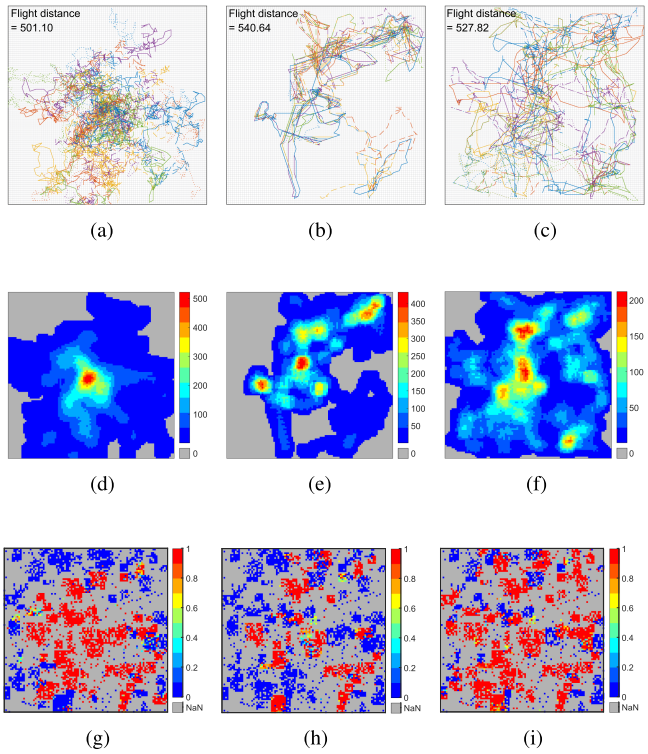

Fig. 11. Visualization of UAV trajectory, visited area, and service rate of 100 × 100 cells using *Enterprise-Network3* data. (a) Trajectory of *LocalGreedyW*. (b) Trajectory of *Ours (w/o neg)*. (c) Trajectory of *Ours*. (d) Visited count of *LocalGreedyW*. (e) Visited count of *Ours (w/o neg)*. (f) Visited count of *Ours*. (g) Service rate of *LocalGreedyW*. (h) Service rate of *Ours (w/o neg)*. (i) Service rate of *Ours*.

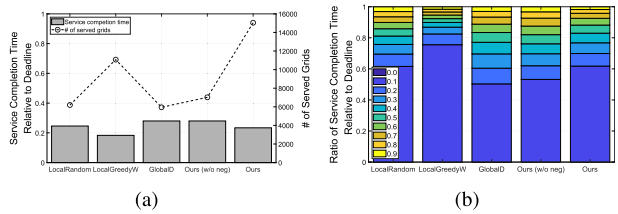

Fig. 12. Performance of served time relative to deadline using *Enterprise-Network3*, with 30 UAVs. (a) Average service completion time per cell. (b) Portion of service completion time of each algorithm.

However, the performance of *Ours (w/o neg)* and *LocalGreedyW* is very similar, implying that without efficient negotiation among UAVs, consideration of long-term strategy becomes ineffective. This means that cooperative targeting is crucial in selecting the targets separately, and our short-term and long-term path planning works well together only when both are applied.

We visualized the cumulative distribution of the average service rate of each traffic heatmap with respect to served traffic ratio in Fig. 10(c). *Ours* showed that a served ratio distribution closer to 1, whereas other counterpart algorithms showed relatively lower served ratio.

We visualize the trajectory and visited area of the UAVs, and the ratio of served traffic relative to total generated traffic of each cell for three best performing algorithms, which are *LocalGreedyW*, *Ours (w/o neg)*, and *Ours* in Fig. 11. The trajectory of each UAV varies in color and line style, and the average travel distance of the UAVs is annotated on the figure. By comparing Fig. 11(a) and (c), we can observe the effect of selecting a global target. Fig. 11(a) shows local movements without moving towards a specific global target, whereas Fig. 11(c) shows both local and global movements towards specific global targets. When we take a closer look at Fig. 11(b) and (c), we can see that Fig. 11(b) shows some duplicate paths of UAVs due to the lack of cooperative targeting, as opposed to Fig. 11(c). The average flight distance for *LocalGreedyW* had the lowest value because the UAVs tend to move locally. On the other hand, *Ours (w/o neg)* and *Ours* had relatively high flight distances with values of 540.64 and 527.82, respectively.

The visit count of each algorithm is visualized in Fig. 11(d), (e), and (f), with different maximum colormap values because of the different trajectory behaviors. This figure shows the locations in which the UAVs visit over a period of 200 timeslots, and the summations of the number of visits per cell. The color of each visited area is mapped to a 10 step color scale with the lowest value of 1 as blue and each maximum value as red, while the grey colored cells indicate 0 visit. Since the traffic request of this *Enterprise-Network3* data occurs in a specific area, Fig. 11(f) shows the visits in the overall area with only 1,055 unvisited cells, whereas Fig. 11(d) and (e) do not show any visits in some part of the RoI, with 2,505 and 4,017 unvisited cells. In Fig. 11(f), *Ours* visited a relatively high number of cells in the areas in which traffic was requested, compared to the blue area of Fig. 11(i). This implies that our path planning algorithm correctly targeted the locations of network traffic.

The cumulative service rate, $\frac{totalServedTraffic}{totalRequestedTraffic}$, per cell is mapped to a 10 step color scale with 0 as blue and 1 as red, while grey colored cells indicate the cells that never requested network traffic. *Ours* shows the largest area of red colored cells, reaching 64.8% of 3,458 cells in which traffic was requested, and the next highest algorithm is *LocalGreedyW*, with only 50.0%. It indicates that our negotiation and scoring system play a key role in traffic service over the area.

We also examine the service completion time relative to the *deadline* of every traffic request, and the number of grid cells served, using *Enterprise-Network3*. The average service completion time $(serviceTime/deadline)$ of each cell over 10 trials is quantified, and the ratio of service completion time shows the portion of each relative completion time. When the metric is high, it indicates that UAVs took a longer time to serve the given traffic requests; and when it is low, it means UAVs took relatively shorter time to serve them. We measure only the service completion time for the served grids, and the value for non-served grids are not counted. In Fig. 12(a), *Ours* showed reasonable service completion time along with the highest number of served grids, serving the average of 15,044 grids, whereas the next highest algorithm, *LocalGreedyW*, served only 11,077 grids. In Fig. 12(b), the ratio of service completion time relative to *deadline* is measured over 10 trials. We can tell that *Ours* serviced almost all grids before half of the given deadline. This result shows that the grids are served only two seconds ago when the $deadline$ is 10 seconds.

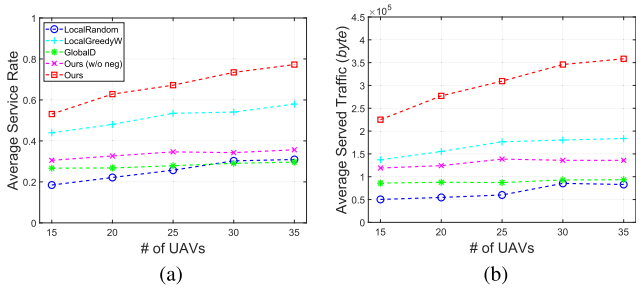

Fig. 13. Average throughput performance with respect to the number of UAVs in *Enterprise-Network3*. (a) Average service rate per cell. (b) Average served traffic in byte.

We examine the effect of the number of UAVs by varying it from 15 to 35. We investigate how the average service rate and the average served traffic are affected, as in Fig. 13. Since *Ours (w/o neg)* and *GlobalD* select the targets without cooperating with neighboring UAVs, there is almost no performance enhancement as the number of UAVs increases. *LocalGreedyW* shows relatively high performance because of its randomness in selecting local targets when the score of all grid points in local area is zero, leading to the spread of UAVs. It implies that our algorithm sustains even when more UAVs are used in the mission, and performance gets improved as more UAVs are engaged, while other algorithms show slight enhancement.

We next investigate the performance of algorithms using different datasets as in Fig. 14. We describe total traffic and total number of cells generating traffic in each dataset. *Ours* shows high performance for all datasets, whereas the performance of the other algorithms varies depending on dataset. In *Enterprise-Network1*, *GlobalD*, and *Ours (w/o neg)* show higher performance than *LocalGreedyW* because there are few grouped traffic generated in one traffic heatmap, which makes it hard for *LocalGreedyW* to find traffic, while *GlobalD* and *Ours (w/o neg)* estimates the location of traffic by checking future values. The relative performance of *GlobalD* and *Ours (w/o neg)* decreases when there are more spotted, scattered traffic requests. When there is no cooperative targeting, UAVs will ignore other groups of traffic. In Fig. 14(c), *Ours (w/o neg)* shows slightly better performance than *GlobalD*, implying that considering local target is essential when there is a large amount of $Local$ or $Spot$ traffic rather than few $Heavy$ traffic.

Using the simulation datasets, we validate the proposed scheme part by part, by comparing *Ours* with other algorithms that uses only some parts of the scheme. The overall results imply that *Ours* is effective when all of necessary steps are integrated. Other algorithms, *Ours (w/o neg)*, *GlobalD*, and *LocalGreedyW* show relatively lower performance than *Ours* since they do not fully consider each critical step.

C. Real-World Dataset-Based Validation

We validate performance of our proposed scheme by comparing it with well-known path planning algorithms such as *Random Waypoint* called *RWP* [39], m-TSP genetic algorithm called *mTSP-GA* [40], *Time Routing* [41], and *Weighted Sum* [42], [43]. For the *RWP* algorithm, we set the walk interval to zero to one timeslot, and the pause interval to zero to five timeslots. Since *mTSP-GA* requires a set of targets to visit, we selected grid points that have traffic requests in the global area as targets. We modified *mTSP-GA* to select a target at every *window* timeslots, and generate efficient paths to tour targets. In *mTSP-GA*, in case that the window period has not been expired yet after the tour, the UAV stays in the final target. If the window period is over during the tour, it starts selecting a new target. Also, we slightly modified the proposed algorithm called *Time Routing* [41]. *Time Routing* algorithm first divides the RoI into several sub areas using the k-means clustering, and uses a distance based weighted sum method to select the candidate points to visit. After selecting the candidates, UAV calculates utility metric for all possible paths. Lastly, for *WeightedSum*, we modified the traditional minimum weighted sum algorithm to select the best grid point, considering the deadline of traffic request at the cells and the distance to the grid point. The score of each grid point in the local area is calculated as follows:

$$score(G_i) = \alpha \cdot \Sigma_{c \in coveredCell(G_i)} D_c \cdot T_c + (1-\alpha) \cdot 1/d(G_{current}, G_i) \cdot \beta, \quad (6)$$

where $G_{current}$ is the current location of the UAV, G_i is a candidate grid point i that UAV can visit within one timeslot, $coverableCellList(G_i)$ is the list of cells that a UAV can cover when the UAV visits G_i. D_c is the age of the network traffic, which indicates urgency among remaining traffic, and T_c indicates the amount of traffic generated in cell, $d(G_{current}, G_i)$ is the distance from the current location to grid point i, and α and β are tuning parameters. The UAV selects a grid point that has the highest score. We additionally compare *Ours* with the static UAV situation, namely *Static*, in which UAVs randomly distributed serve network traffic without movements.

We compare our algorithm with general path planning algorithms using a real-world dataset. The real-world dataset *City-Cellular-Network* [11] used in the experiments includes cellular traffic records in the city area of China, which are collected for eight days from August 19 to August 26, 2012 every hour, for a total of 192 hours, from 13,296 base stations. The dataset includes the relative location of each base station. Since our experiment requires cell based traffic values and square-shaped RoIs, we crop the original area and produce a square-shaped RoI. The original 192 hours are mapped to 192 ts with one ts period identical to our environment setting. Since the relative location of the base stations made it hard to identify the real location and real size of the total area, we followed the relative position of base stations and the flow of traffic without considering real locations of base stations. We adjusted the maximum traffic value over time and space to 6,000 because the original data is traffic requested for base stations. As we consider the situation of sudden traffic requests that ground-based stations can not handle, we set a baseline of served traffic to 1,500 to 2,500 for each cell. We named the real-world datasets as *City-Cellular-Network1* with a baseline of 2,500, *City-Cellular-Network2* with a baseline of 2,000, and *City-Cellular-Network3* with a baseline of 1,500. *City-Cellular-Network2* is used in our main real-world

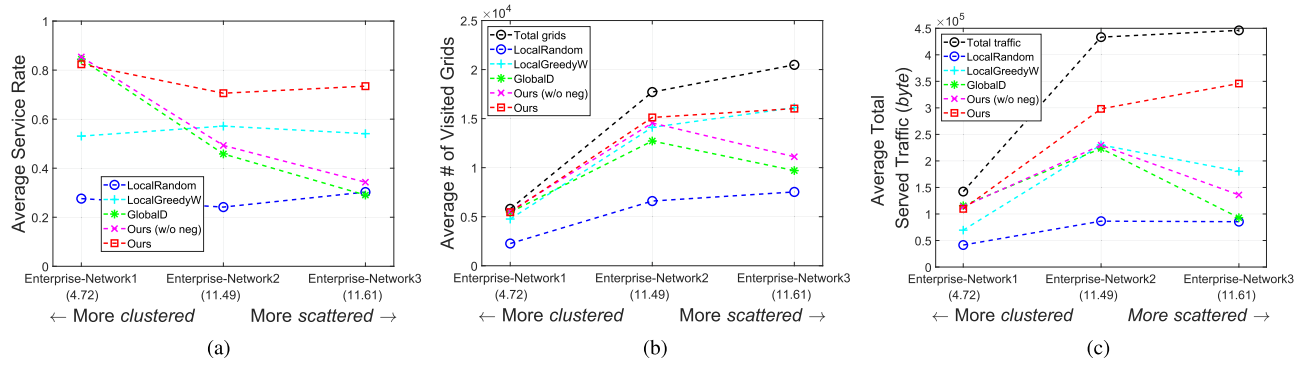

Fig. 14. Performance with regard to simulation dataset using 30 UAVs. (a) Average service rate per cell. (b) Average number of visited grids. (c) The average served traffic in byte.

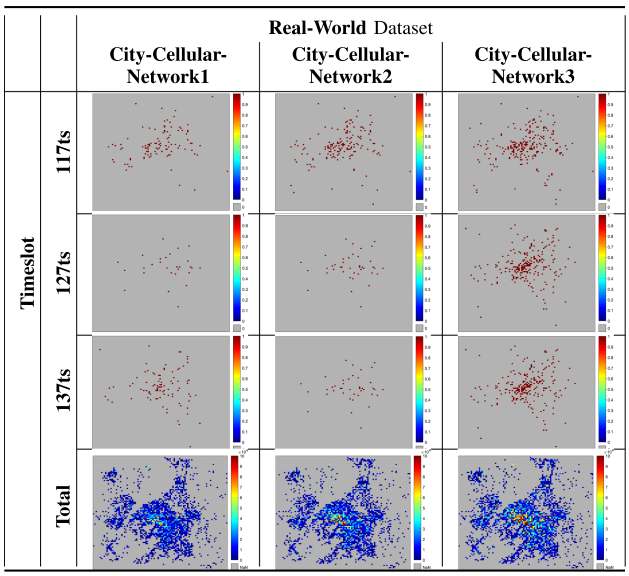

Fig. 15. Visualization of the real world datasets, *City-Cellular-Network1*, *City-Cellular-Network2*, and *City-Cellular-Network3* at selected timeslots.

experiments. The traffic flow and total traffic requests are visualized in Fig. 15. As the real-world traffic requests are concentrated in the city area, the amount of traffic requests is relatively higher in the center area than other border areas. We have modified the limit to emphasize the difference in total traffic request.

We first evaluate the amount of served traffic as in Fig. 16(a) and (b), using *City-Cellular-Network2*. Since the network traffic request is scattered over the RoI, even *Ours* failed to perfectly serve all traffic in a timeslot. However, *Ours* showed the highest served amount in Fig. 16(a), while the second highest changed over time, from *mTSP-GA* to *WeightedSum*. In Fig. 16(b), *Ours* showed the highest served amount of 16, 804, 800 bytes, which is 67% of the total traffic requested, while the next highest algorithm served only 58%. We also evaluate the average served rate per timeslot in Fig. 16(c). *Ours* showed the highest average served rate, with the highest lower bound value of 0.47, while others showed values below 0.3. It indicates that at least 50% of the total traffic requests in the timeslot are served by our algorithm, while other general schemes only served 30% of total requests.

We visualize the trajectory of the UAVs, the areas visited by UAVs, and the rate of network traffic served over the RoI of top four algorithms of *WeightedSum*, *mTSP-GA*, *Time Routing*, and *Ours* as in Fig. 17. Because of the characteristics of *City-Cellular-Network2*, the traffic requests occur mostly at the center area of the RoI. The trajectories shown in Fig. 17(j) show that UAVs move around the central area. In Fig. 17(a), the movement paths of UAVs are overlapped each other compared against Fig. 17(j). In Fig. 17(d), the UAVs moved globally compared to other algorithms because *mTSP-GA* has selected a global target until there is no request in the global area, making UAV to select a global target and move even if there is only one cell that requested traffic. Therefore, even though *mTSP-GA* visited a broad area, it failed to serve the essential heavy traffic requests occurred in the center region (Fig. 17(f)). In Fig. 17(g), as each UAV has a designated area, UAVs first move towards to their own area along with the global trajectory, and then plan their own path within the area. In Fig. 17(l), *Ours* shows larger red area than any other algorithms, while targeting relatively higher traffic located near the center area. The average flight distance of each algorithm is annotated in the above figures. *WeightedSum* shows the lowest value of 270.76 m, while *mTSP-GA*, *Time Routing* and *Ours* show 559.39 m, 511.10 m and 550.05 m, respectively. *Time Routing* shows lower travel distance than *Ours* since *Time Routing* takes into account distance with priority. However, as heavy traffic requests occur mainly in the center area (Fig. 15), the blue part in Fig. 17(i) indicates that the total amount of served traffic is lower than *Ours*. This observation implies that our global target selectively handles heavy traffic and the cooperative targeting scheme accurately selects the next essential location to serve.

We observe the relative service completion time relative to deadline and throughput in terms of the number of served grids in Fig. 18. In Fig. 18(a), *Ours* shows the highest number of served grids and the lowest average service completion time with a value of 0.181, meaning that the traffic requests are served at the time of 18.1% out of the given deadline. The next lowest algorithm, *WeightedSum* shows the relative service completion time of 22% and a lower number of grid squares served. *mTSP-GA* has a relatively high service completion time due to inconsideration of the deadline in network traffic. *Time Routing* shows high completion time compared to other algorithms, along with small

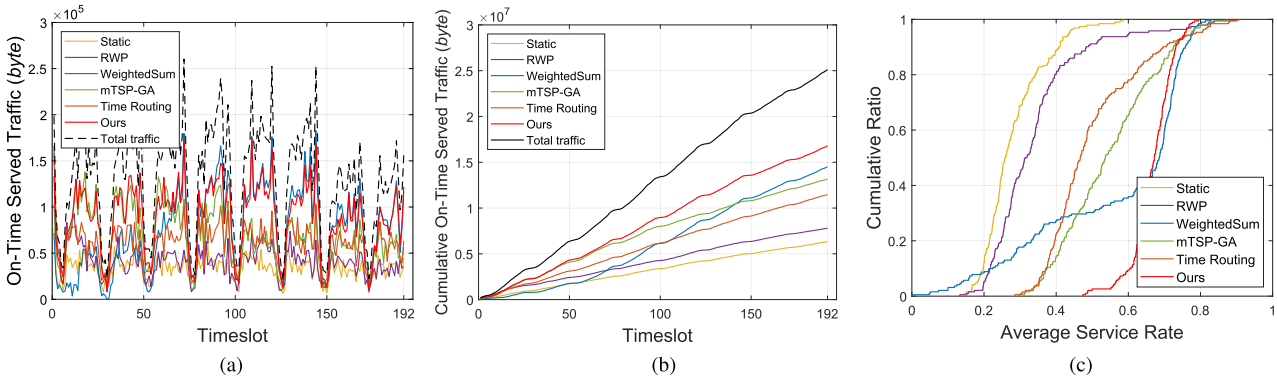

Fig. 16. Performance dynamics of served traffic amount and cumulative distribution of served rate of each timeslot with 35 UAVs using *City-Cellular-Network2*. (a) Amount of served traffic per heatmap. (b) Cumulative growth of served traffic. (c) Cumulative ratio of average service rate.

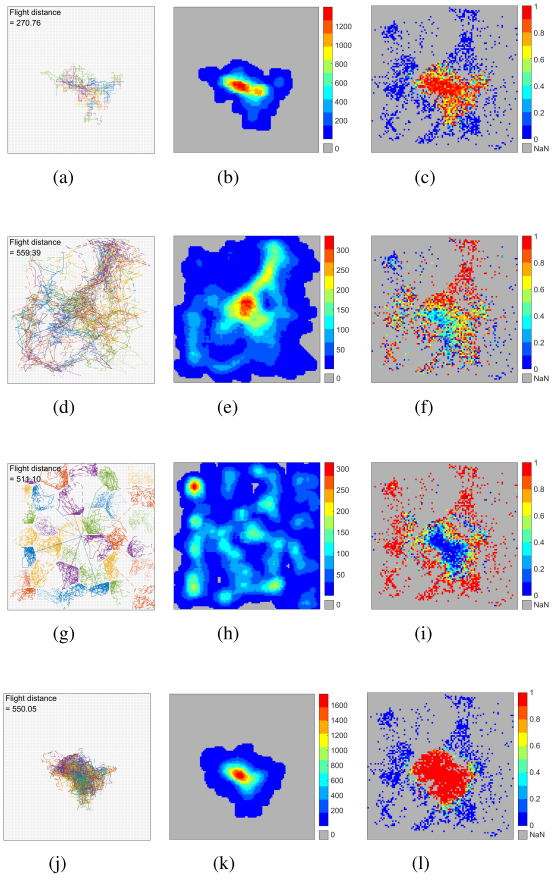

Fig. 17. Visualization of UAV trajectory, visited area, and traffic service rate of 100×100 cells using *City-Cellular-Network2*. (a) Trajectory of *WeightedSum*. (b) Visited count of *WeightedSum*. (c) Service rate of *WeightedSum*. (d) Trajectory of *mTSP-GA*. (e) Visited count of *mTSP-GA*. (f) Service rate of *mTSP-GA*. (g) Trajectory of *Time Routing*. (h) Visited count of *Time Routing*. (i) Service rate of *Time Routing*. (j) Trajectory of *Ours*. (k) Visited count of *Ours*. (l) Service rate of *Ours*.

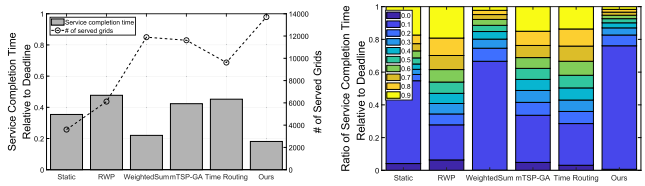

Fig. 18. Performance of served time relative to deadline using *City-Cellular-Network2*, with 35 UAVs. (a) Average service completion time per cell. (b) Portion of service completion time of each algorithm.

number of served grids. In Fig. 18(b), *Static* shows a relatively high portion of 90%, compared to 80% to 20% for the other algorithms, which means there are some cells with network traffic requests that a single UAV can only serve by visiting the same grid 10 times. *Ours* shows the largest portion for 10% time completion compared to other algorithms, with almost 90% of grids are served before half of the deadline. This result implies that *Ours* effectively serves many grids in reasonable service completion time compared to other algorithms.

We compared performance in terms of average service rate, the number of visited grids, total served traffic in bytes based on various datasets of *City-Cellular-Network1*, *City-Cellular-Network2*, and *City-Cellular-Network3* as in Fig. 19. In Fig. 19(b) and (c), the amount of network traffic requested by grids and the total sum of network traffic requests increased from *City-Cellular-Network1* to *City-Cellular-Network3*. Therefore, the performance of all algorithms except *Ours* shows a significant decrease, whereas *Ours* shows the most stable performance. Since *WeightedSum* does not include cooperative target selection, when there are more traffic requests and the score of a certain grid point is high, the performance has degraded by keeping selecting the same grids. However, *mTSP-GA* and *Time Routing* show relatively stable performance than *WeightedSum*, since *Time Routing* divides the area and *mTSP-GA* has some randomness when selecting the path. This result implies that *Ours* can survive in the traffic congestion scenario.

We additionally compared our group UAV job assignment strategy with three different methods using two UAVs as in Fig. 20. We implemented an algorithm named *LocalOpt* that checks all of the future requests one by one and selects the optimal assignment with regard to served traffic in an ideal and brute-force manner; *Dist-ver1* that assigns a UAV with a closer distance to the global target as *ReverseUAV*; and *Dist-ver2* that assigns a UAV with a farther distance to the global target. Since the random assignment does not show any noticeable difference

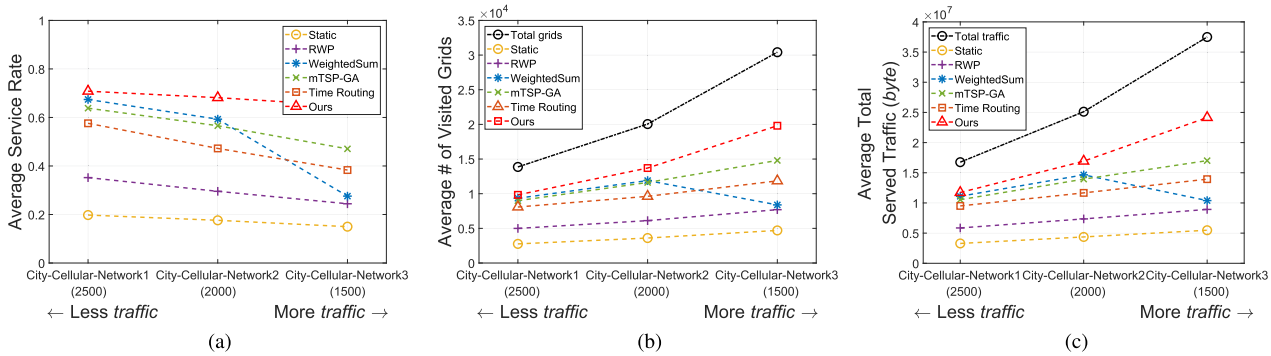

Fig. 19. Performance with regard to the real-world dataset using 35 UAVs. (a) Average service rate per cell. (b) Average number of visited grids. (c) The average served traffic in byte.

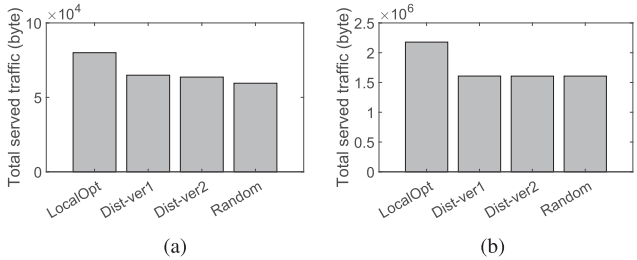

Fig. 20. Performance with respect to group UAV assignment strategy. (a) *Enterprise-Network3*. (b) *City-Cellular-Network2*.

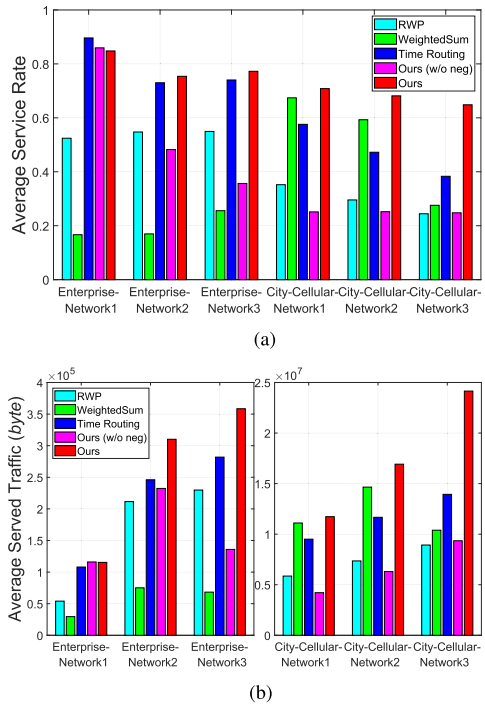

Fig. 21. Performance of algorithms with respect to the dataset using 35 UAVs with capacity of 4,000 bytes. (a) Average service rate per cell. (b) Average served traffic in byte.

in performance compared to other assignment strategies, we take the simple yet efficient random assignment as our main strategy.

We finally evaluate performance in terms of average service rate and average served traffic across all datasets using 35 UAVs with 4,000 bytes of capacity (Fig. 21). Since the prior simulation data shows relatively fewer traffic requests than real-world data, the net performance is better than that of real-world data. While *Ours* shows stable performance across all datasets, other algorithms show different performances on simulated data and real-world data. As our suggested algorithm aims to serve as much traffic request as possible, the gap between other algorithms is higher in Fig. 21(b) than in Fig. 21(a). *Time Routing* seems to work fine in the prior simulated datasets, but the amount of served traffic shows a huge gap. Moreover, *Time Routing* fails when using real-world datasets, in case of high intensity of traffic requests.

The real-world dataset-based results imply that our scheme copes well with the spatial and temporal fluctuation of real-world network traffic requests compared to other counterpart algorithms. Overall, we have validated that our proposed algorithm successfully serves the most amount of network traffic in a reasonable time.

VII. Conclusion

We have presented a new lightweight yet effective algorithm for network traffic coverage using multiple UAVs for a UAV-enabled network. We have developed a distributed trajectory design for UAVs with two phases of short-term path planning and long-term path planning, to serve continuous network traffic requests over time throughout an RoI. We have demonstrated that our algorithm provides a UAV-enabled network with reasonable average service rate and amount of served traffic, while reducing the service time, against some existing counterpart algorithms. Our work takes into account the dynamic network traffic requests over time and space in an RoI, which has rarely been considered in past related works. Our work opens a new perspective that makes UAVs engaged to serve temporally and spatially varying traffic requests along with dynamic mobility of users, rather than just offering services to every static user over time.

For future work, we may extend our scheme from homogeneous multiple UAV-enabled networks to heterogeneous multiple UAV-enabled networks, with various UAV characteristics like computation, communication, and battery capacity. Since the battery usage required for traveling may affect the grouping of UAVs, it would be interesting to take into account the flight

distance of UAVs, to minimize battery usage, UAV failure situations, or balanced trajectory designs for the optimal UAV role assignment. Considering the battery usage, trajectory planning can be extended with recharging movements. Moreover, we can consider some more physical layer characteristics in channel and interference or 3D locations and collisions to be applied to real-world systems for more practical aspects.

REFERENCES

[1] H. Shinbo, Y. Kunisawa, T. Sakai, Y. Kitatsuji, A. Endo, and K. Tanaka, "Flying base station for temporary mobile communications in an area affected by a disaster," in *Proc. IEEE 5th Int. Conf. Inf. Commun. Technol. Disaster Manage.*, 2018, pp. 1–7.

[2] M. D. Nguyen, T. M. Ho, L. B. Le, and A. Girard, "UAV trajectory and sub-channel assignment for UAV based wireless networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2020, pp. 1–6.

[3] E. Montero, D. Rosário, and A. Santos, "Clustering users for the deployment of UAV as base station to improve the quality of the data," in *Proc. IEEE Latin- Amer. Conf. Commun.*, 2019, pp. 1–6.

[4] Google X, "Expanding internet connectivity with stratospheric balloons," 2020. Accessed on: Jan. 07, 2022. [Online]. Available: https://x.company/projects/loon/

[5] B. Galkin, J. Kibilda, and L. A. DaSilva, "Deployment of UAV-mounted access points according to spatial user locations in two-tier cellular networks," in *Proc. IEEE Wireless Days*, 2016, pp. 1–6.

[6] 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.

[7] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.

[8] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Commun. Lett.*, vol. 21, no. 3, pp. 604–607, Mar. 2017.

[9] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, Jun. 2018.

[10] Z. Qin, Z. Liu, G. Han, C. Lin, L. Guo, and L. Xie, "Distributed UAV-BSS trajectory optimization for user-level fair communication service with multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 70, no. 12, pp. 12290–12301, Dec. 2021.

[11] X. Chen, Y. Jin, S. Qiang, W. Hu, and K. Jiang, "Analyzing and modeling spatio-temporal dependence of cellular traffic at city scale," in *Proc. IEEE Int. Conf. Commun.*, 2015, pp. 3585–3591.

[12] G. Barlacchi et al., "A multi-source dataset of urban life in the city of Milan and the province of Trentino," *Sci. Data*, vol. 2, no. 1, pp. 1–15, 2015.

[13] X. Wang et al., "Spatio-temporal analysis and prediction of cellular traffic in metropolis," *IEEE Trans. Mobile Comput.*, vol. 18, no. 9, pp. 2190–2202, Sep. 2019.

[14] S. P. Sone, J. J. Lehtomäki, and Z. Khan, "Wireless traffic usage forecasting using real enterprise network data: Analysis and methods," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 777–797, 2020.

[15] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices (extended version)," *IEEE/ACM Trans. Netw.*, vol. 20, no. 3, pp. 662–676, Jun. 2012.

[16] J. Wang et al., "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[17] X. Sun and N. Ansari, "Jointly optimizing drone-mounted base station placement and user association in heterogeneous networks," in *Proc. IEEE Int. Conf. Commun.*, 2018, pp. 1–6.

[18] J. Qin, Z. Wei, C. Qiu, and Z. Feng, "Edge-prior placement algorithm for UAV-mounted base stations," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2019, pp. 1–6.

[19] Q. Zhang, W. Saad, M. Bennis, X. Lu, M. Debbah, and W. Zuo, "Predictive deployment of UAV base stations in wireless networks: Machine learning meets contract theory," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 637–652, Jan. 2021.

[20] F. Cheng et al., "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.

[21] N. Zhao et al., "Joint trajectory and precoding optimization for UAV-assisted NOMA networks," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3723–3735, May 2019.

[22] H. Y. Alsheyab, E. Bedeer, S. Choudhury, and S. Ikki, "Multi-UAV wireless networks: Jointly trajectory optimization and resource allocation," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2022, pp. 1053–1058.

[23] Y. Qian, K. Sheng, C. Ma, J. Li, M. Ding, and M. Hassan, "Path planning for the dynamic UAV-aided wireless systems using Monte Carlo tree search," *IEEE Trans. Veh. Technol.*, vol. 71, no. 6, pp. 6716–6721, Jun. 2022.

[24] S. Javed, A. Hassan, R. Ahmad, W. Ahmed, M. M. Alam, and J. J. Rodrigues, "UAV trajectory planning for disaster scenarios," *Veh. Commun.*, vol. 39, 2023, Art. no. 100568. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214209622001152

[25] W. Tian, X. Ding, G. Liu, Y. Dai, and Z. Han, "A UAV-assisted secure communication system by jointly optimizing transmit power and trajectory in the Internet of Things," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 4, pp. 2025–2037, Dec. 2023.

[26] J. Lee and V. Friderikos, "Multiple UAVs trajectory optimization in multi-cell networks with adjustable overlapping coverage," *IEEE Internet Things J.*, vol. 10, no. 10, pp. 9122–9135, May 2023.

[27] C. Hao, Y. Chen, Z. Mai, G. Chen, and M. Yang, "Joint optimization on trajectory, transmission and time for effective data acquisition in UAV-enabled IoT," *IEEE Trans. Veh. Technol.*, vol. 71, no. 7, pp. 7371–7384, Jul. 2022.

[28] N. Lin, Y. Fan, L. Zhao, X. Li, and M. Guizani, "GREEN: A global energy efficiency maximization strategy for multi-UAV enabled communication systems," *IEEE Trans. Mobile Comput.*, vol. 22, no. 12, pp. 7104–7120, Dec. 2023.

[29] H. Huang and A. V. Savkin, "Deployment of heterogeneous UAV base stations for optimal quality of coverage," *IEEE Internet Things J.*, vol. 9, no. 17, pp. 16429–16437, Sep. 2022.

[30] K. K. Nguyen, T. Q. Duong, T. Do-Duy, H. Claussen, and L. Hanzo, "3D UAV trajectory and data collection optimisation via deep reinforcement learning," *IEEE Trans. Commun.*, vol. 70, no. 4, pp. 2358–2371, Apr. 2022.

[31] S. Fu, M. Zhang, M. Liu, C. Chen, and F. R. Yu, "Towards energy-efficient UAV-assisted wireless networks using an artificial intelligence approach," *IEEE Wireless Commun.*, vol. 29, no. 5, pp. 77–83, Oct. 2022.

[32] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 2109–2121, Mar. 2018.

[33] Z. Wang, W. Xu, D. Yang, and J. Lin, "Joint trajectory optimization and user scheduling for rotary-wing UAV-enabled wireless powered communication networks," *IEEE Access*, vol. 7, pp. 181369–181380, 2019.

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

[35] M. R. Brust and B. M. Strimbu, "A networked swarm model for UAV deployment in the assessment of forest environments," in *Proc. IEEE 10th Int. Conf. Intell. Sensors, Sensor Netw. Inf. Process.*, 2015, pp. 1–6.

[36] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A mobility model for UAV ad hoc network," in *Proc. IEEE Int. Conf. Unmanned Aircr. Syst.*, 2014, pp. 383–388.

[37] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Netw.*, vol. 27, pp. 3377–3393, Nov. 2021, doi: 10.1007/s11276-019-02200-6.

[38] F. Xu et al., "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE Trans. Serv. Comput.*, vol. 9, no. 5, pp. 796–805, Sep./Oct. 2016.

[39] M. Boutin, "Random waypoint mobility model," 2022. [Online]. Available: https://kr.mathworks.com/matlabcentral/fileexchange/30939-random-waypoint-mobility-model

[40] J. Kirk, "Multiple traveling salesmen problem - genetic algorithm," 2014. [Online]. Available: https://www.mathworks.com/matlabcentral/mlc-downloads/downloads/submissions/21299/versions/6/previews/mtspf_ga.m/index.html

[41] J. Yoon, S. Doh, O. Gnawali, and H. Lee, "Time-dependent ad-hoc routing structure for delivering delay-sensitive data using UAVs," *IEEE Access*, vol. 8, pp. 36322–36336, 2020.

[42] A. A. Somasundara, A. Ramamoorthy, and M.B. Srivastava, "Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines," in *Proc. IEEE 25th Int. Real-Time Syst. Symp.*, 2004, pp. 296–305.

[43] J. Yoon, Y. Jin, N. Batsoyol, and H. Lee, "Adaptive path planning of UAVs for delivering delay-sensitive information to ad-hoc nodes," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2017, pp. 1–6.

JeiHee Cho received the B.S. degree in computer science and engineering in 2021 from Ewha Womans University, Seoul, South Korea, where she is currently working toward the M.S. degree in computer science and engineering. Her research interests include network security, VANET, machine learning, and artificial intelligence-driven network system design.

SooMin Ki is currently working toward the B.S. degree with the Division of Brain and Cognitive Sciences, Department of Computer Science and Engineering, Ewha Womans University, Seoul, South Korea. Her research interests include the IoT, networked systems, wireless *ad hoc* networks, and deep learning.

HyungJune Lee (Member, IEEE) received the B.S. degree in electrical engineering from Seoul National University, Seoul, South Korea, in 2001, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2006 and 2010, respectively. He joined Broadcom as a Senior Staff Scientist for working on research and development of 60 GHz 802.11ad SoC MAC. He also worked with AT&T Labs as a Principal Member of Technical Staff with the involvement of LTE overload estimation, LTE-WiFi interworking, and heterogeneous networks. He is currently an Associate Professor with the Computer Science and Engineering Department, Ewha Womans University, Seoul. His research interests include distributed learning and future wireless networks on the IoT, fog computing, VANET, and machine learning-driven network system design.