Dual Time-Constrained UAV Routing for Reward Maximization Under Joint Multi-PoI Capture

Chun-An Yang[§‖], Guo-Wei Huang[†‖], Kuan-Hsiang Lo[†], Jian-Jhih Kuo[†*], Yu-Wen Chen[‡], and Ming-Jer Tsai[§]
[§]Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
[†]Dept. of Computer Science and Information Engineering, National Chung Cheng University, Chiayi, Taiwan
[‡]Computer Systems Technology, New York City College of Technology, Brooklyn, New York, USA

***Abstract*—Unmanned aerial vehicles (UAVs) are widely used in various applications, such as surveillance and disaster response. These missions often involve video recording of multiple spatially distributed points-of-interest (PoIs). However, limited battery capacity imposes a strict dual time constraint, making efficient route planning essential. A green effective strategy is to capture multiple PoIs simultaneously from a single capture point, where the required capture time at that point is determined by the longest duration among the captured PoIs. Moreover, PoIs may carry different reward values that reflect their priority or importance in the mission. Therefore, it is critical to jointly select the PoIs to be captured and their associated capture points, such that the UAV visits each selected point within the tour and capture time budget while maximizing the total collected reward. The paper formulates an optimization problem and proposes an approximation algorithm. Finally, simulation results demonstrate that our algorithm outperforms other baselines by up to 67%.**

I. Introduction

Unmanned aerial vehicles (UAVs) have gained increasing prominence due to their agility, remote operability, and rapid deployment capabilities. They play vital roles in various applications [1]–[6], including data collection, parcel delivery, image surveillance, and disaster response. In scenarios where roads collapse or become inaccessible, UAVs provide flexible and scalable solutions that traditional vehicles cannot [5]. They can access remote or hazardous areas and transmit real-time visual or sensor data to support rescue teams [2]. Despite their advantages, UAVs are constrained by limited battery capacity, which restricts their mission duration [1]–[6].

Usually, points-of-interest (PoIs) refer to collapsed buildings, areas where individuals may be trapped, or priority locations for aid distribution [4], [5]. Command centers must obtain timely aerial video footage of these critical sites to assess damage and coordinate rescue efforts. However, if a UAV attempts to inspect each PoI independently, its route may become excessively long. Given the constraint of limited tour time, a green efficient approach is to capture multiple PoIs simultaneously from a single location [4], [6]. In other words, each PoI can be captured from a subset of potential capture points, and these subsets may overlap. Additionally, each PoI requires a specific recording duration to meet its requirement [3]. When multiple PoIs are covered from the same location, the UAV does not need to record them separately. Instead, the capture time at that location is determined by the PoI that requires the longest duration, efficiently saving capture cost.

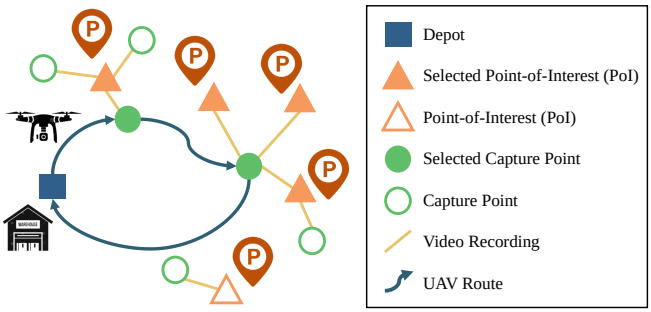


Fig. 1. UAV routing illustration for video recording.

Moreover, each PoI contributes a certain level of reward that reflects its priority or importance in the mission [2], [5]. However, due to the limited tour and capture time budget, UAV cannot capture all PoIs within a single tour. To ensure that mission resources are allocated to the most valuable targets, it is crucial to properly select a subset of capture points visited by the UAV that cover multiple PoIs while maximizing the total reward. Note that determining which PoIs should be captured at a capture point is essential, as this decision will affect the capture time at that location. For clarity, Fig. 1 illustrates a UAV routing, where the UAV departs from a depot, visits selected capture points covering PoIs, and returns to the depot.

Optimizing the total reward of this problem involves three challenges: 1) *Reward versus time cost*. A PoI with a higher reward may incur greater tour and capture time costs. Under a limited time budget, it is crucial to strike a balance between reward and cost—whether to pursue higher rewards at the expense of greater resource consumption, or to conserve resources to select a larger number of lower-reward but less costly PoIs. 2) *Capture point selection*. A single capture point may capture multiple PoIs, with its capture time determined by the PoI requiring the longest duration. Capturing multiple PoIs simultaneously can help reduce costs and increase the total reward. However, visiting such a point may involve traveling long distances, resulting in a longer tour time despite a shorter capture time. Therefore, it is essential to carefully select capture points by jointly considering reward, tour time, and capture time to maximize the overall reward within the time budget. 3) *UAV tour planning*. Proper tour planning is important, as it significantly impacts the tour cost. Overall, these factors make the problem both complex and challenging.

‖: equal contributions; *: corresponding author (lajacky@cs.ccu.edu.tw)
This work was supported in part by the National Science and Technology Council, Taiwan, under Grants 113-2221-E-194-040-MY3, 114-2221-E-007-079-, and 114-2628-E-194-002-MY3.

To address these challenges, we formulate an optimization problem, termed Dual **T**ime-Constrained **U**AV **R**outing **P**roblem (TURP), which asks for a single UAV tour that departs from a depot, visits a subset of capture points covering partial PoIs, and returns to the depot. The goal is to maximize the total reward while adhering to the limited tour and capture time budget. To solve the TURP, we propose an approximation algorithm, called Dual **T**ime-Constrained **U**AV **R**outing **A**lgorithm (TURA), consisting of three phases. Simulation results show that the TURA outperforms other baselines by up to 67%.

II. Related Work

The applications of UAVs (or mobile chargers) have become increasingly prevalent in a variety of contexts. There are many studies working on route planning for data collection and sensor charging. Zhang *et al*. [2] designed an evolutionary algorithm to determine multiple UAV routes from sources to destinations such that the rewards of data collection are maximized. Liang *et al*. [7] presented a $\frac{1}{4}$-approximation algorithm to determine a mobile charger tour starting and ending at the depot, with the goal of maximizing the charging rewards. Xu *et al*. [5] proposed a $\frac{1}{3}$-approximation algorithm to determine multiple UAV tours starting and ending at the depot, with the goal of maximizing the total monitoring rewards in disaster areas. However, the above studies assume that each PoI has a fixed capture location and that the UAV must visit these locations to maximize the collected rewards.

The following studies further consider a single location can capture/charge multiple PoIs (or sensors) simultaneously. Ma *et al*. [8] presented a $\frac{1}{2}(1-\frac{1}{e})$-approximation algorithm that selects a subset of sensors to cover nearby sensors and plans a charging tour. The goal was to maximize the charging utility, subject to an energy capacity and tour constraint. However, this work neglects the tour cost in the ratio and does not consider the domination of the longest charging/capture time. Kuo *et al*. [9] studied the problem of selecting a subset of connected routers that cover multiple users so as to maximize a reward function under a router node budget constraint. They proposed a $(D_{max}+1)O(\sqrt{B})$-approximation algorithm, where B and D_{max} denote the node budget and the maximum degree of the graph. D'Angelo *et al*. [10] provided an approximation ratio of $(O(\frac{\sqrt{B}}{\epsilon^3}), 1+\epsilon)$ for the rooted version of the problem. Nevertheless, our setting involves the depot, the heterogeneous tour time costs across different edges as well as the capture cost. In addition, we account for the domination of the longest capture time, which further complicates the TURP.

III. The Optimization Problem – TURP

The network considers a set of PoIs $P = \{p_1, p_2, \ldots, p_{|P|}\}$, each PoI $p \in P$ can be captured at a subset of neighboring capture points, denoted as S_p. The complete set of capture points is defined as $S = \{s_0\} \cup_{p\in P} S_p$, where s_0 represents the depot. Note that the subsets S_p may overlap, indicating that multiple PoIs can share common capture point. The PoI network is defined as $G = (V, E)$, where the node set $V = (P \cup S)$ and the edge set $E = (E^{tr} \cup E^{ca})$. The set E^{tr} consists of traveling edges, where each edge (i, j) connects two capture points $i, j \in S$ with an associated travel time cost $t^{tr}_{ij} \geq 0$, which is proportional to the distance. The set E^{ca} consists of capturing edges, where each edge (p, i) connects a PoI $p \in P$ to a capture point $i \in S$ with a capture time cost $t^{ca}_{pi} > 0$. Each PoI has a specific capture time by the UAV and can only be captured at its neighboring points.[1] The capture time t^{ca}_{pi} is a non-zero value when $i \in S_p$. Otherwise, it is set to a positive infinity (i.e., $t^{ca}_{pi} = +\infty$ when $i \notin S_p$).

The benefit of selected PoIs is evaluated with a monotone submodular reward function $R : 2^P \to \mathbb{R}^+$, which satisfies the following two properties [12]: (i) $R(\mathcal{A}) \leq R(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq P$; (ii) $R(\mathcal{A} \cup \{p\}) - R(\mathcal{A}) \geq R(\mathcal{B} \cup \{p\}) - R(\mathcal{B}), \forall \mathcal{A} \subseteq \mathcal{B} \subseteq P, p \in P \setminus \mathcal{B}$. The Dual **T**ime-Constrained **U**AV **R**outing **P**roblem (TURP) focuses on optimizing the UAV tour, which departs from the depot, selectively visits a subset of capture points associated with PoIs, and returns to the depot. The objective is to maximize the total reward obtained from capturing PoIs, subject to the constraint that the total tour and capture time remains within the given time budget B.

For ease of presentation, in this formulation, we consider the reward function as $R(\mathcal{P}) = \sum_{p\in\mathcal{P}} r_p$, where r_p denotes the individual reward of each PoI p and $\mathcal{P} \subseteq P$.[2] To formulate the problem, we introduce three decision variables. First, the binary variable x_{ij} is set to 1 if the UAV traverses edge $(i, j) \in E^{tr}$ and 0 otherwise. Second, the binary variable y_{pi} indicates whether each PoI $p \in P$ is captured at capture point $i \in S$; that is, $y_{pi} = 1$ if p is captured there and $y_{pi} = 0$ otherwise. Finally, the continuous variable $t_i \geq 0$ represents the UAV's capture time at point $i \in S$. The TURP can be modeled as a mixed-integer linear programming (MILP) as follows.

$$\max \quad \sum_{p\in P}\sum_{i\in S} r_p \cdot y_{pi} \tag{1a}$$

$$\text{s.t.} \quad \sum_{i\in S\setminus\{s_0\}} x_{s_0 i} = 1 \tag{1b}$$

$$\sum_{i\in S\setminus\{s_0\}} x_{i s_0} = 1 \tag{1c}$$

$$\sum_{j\in S\setminus\{i\}} x_{ij} \leq 1, \qquad \forall i \in S \tag{1d}$$

$$\sum_{j\in S\setminus\{i\}} x_{ji} \leq 1, \qquad \forall i \in S \tag{1e}$$

$$\sum_{j\in S\setminus\{i\}} x_{ij} \geq y_{pi}, \qquad \forall i \in S \setminus \{s_0\}, \forall p \in P \tag{1f}$$

$$\sum_{j\in S\setminus\{i\}} x_{ji} \geq y_{pi}, \qquad \forall i \in S \setminus \{s_0\}, \forall p \in P \tag{1g}$$

$$\sum_{k\in S'}\sum_{j\in S\setminus S'} x_{kj} \geq y_{pi}, \forall S' \subseteq S \setminus \{s_0\}, \forall i \in S', \forall p \in P \tag{1h}$$

$$\sum_{k\in S'}\sum_{j\in S\setminus S'} x_{jk} \geq y_{pi}, \forall S' \subseteq S \setminus \{s_0\}, \forall i \in S', \forall p \in P \tag{1i}$$

[1]The video for a PoI must be continuously captured without interruption, especially during rescue missions, due to factors such as information integrity, scene coherence, technical considerations, and operational efficiency [11].

[2]This formulation can be easily extended to the formulation of general submodular functions.

$$\sum_{i \in S \setminus \{s_0\}} y_{pi} \leq 1, \qquad \forall p \in P \tag{1j}$$

$$t^{ca}_{pi} \cdot y_{pi} \leq t_i, \qquad \forall i \in S, \forall p \in P \tag{1k}$$

$$\sum_{(i,j) \in E^{tr}} t^{tr}_{ij} \cdot x_{ij} + \sum_{i \in S} t_i \leq B \tag{1l}$$

$$x_{ij}, y_{pi} \in \{0,1\}, \qquad \forall i,j \in S, \forall p \in P \tag{1m}$$

$$t_i \geq 0, \qquad \forall i \in S \tag{1n}$$

The objective (1a) aims to maximize the total reward from capturing PoIs. The constraints (1b) and (1c) make sure that the UAV route starts and ends at the depot s_0. The constraints (1d) and (1e) limit the flow into and out of capture point i at most 1. The constraints (1f) and (1g) ensure that capture point i must be followed and preceded by exactly one other capture point j if point i is selected for PoI p. The constraints (1h) and (1i) further guarantee that the UAV route forms a single tour starting and ending at the depot (or contains a path connecting each PoI and the depot). The constraint (1j) limits that at most one capture point is selected for each PoI p. The constraint (1k) decides the capture time of each capture point i depending on the longest time t^{ca}_{pi} of PoI p captured at point i. Finally, the constraint (1l) limits the UAV's tour and capture time.

Note that the TURP is NP-hard since the maximum submodular set function problem [12] can be reduced to it. Due to the page limit, the proof is detailed in Appendix A of [13].

Theorem 1. The TURP is NP-hard.

IV. Algorithm Design — TURA

To solve the TURP efficiently, we propose an approximation algorithm named Dual **T**ime-Constrained **U**AV **R**outing **A**lgorithm (TURA) to simultaneously determine a subset of PoIs to be captured, select their corresponding capture points, and generate a UAV tour visiting these selected capture points within the given tour and capture time budget. To deal with joint multi-PoI capture, we first construct *a directed auxiliary graph (DAG)* $A = (V \cup V', E \cup E')$ based on the PoI network $G = (V, E)$, where V' and E' denote the sets of virtual nodes and edges. The DAG A connects two or more PoIs sharing the same capture point via the assistance of the virtual nodes and edges. To determine the captured PoIs and their capture points, we iteratively select a capture point guided by the *marginal gain* and construct *a directed reference tree (DRT)*, rooted at the depot, that spans all selected PoIs through their associated capture points in A. To derive a UAV tour, the final DRT T is pruned to a subtree T' by retaining only the depot, the selected capture points, and the travel edges connecting them. A depth-first search (DFS) traversal with shortcutting is then performed on T' to form a tour that visits each selected capture point exactly once before returning to the depot.

The proposed TURA consists of three main phases: 1) Directed Auxiliary Graph Construction (DAGC), 2) PoI and Its Capture Point Selection (PCPS), and 3) UAV Tour Generation (UTG). In the DAGC phase, a DAG is constructed by duplicating the original graph G and introducing additional virtual nodes and edges. The PCPS phase then selects a subset of capture points via the marginal gain and constructs a DRT that spans all selected PoIs through their associated capture points. Finally, the UTG phase prunes the DRT into a subtree, performs a DFS traversal on the pruned tree, and derives a UAV tour. Each phase is detailed in the following subsections, and the algorithm is shown to achieve an approximation ratio.

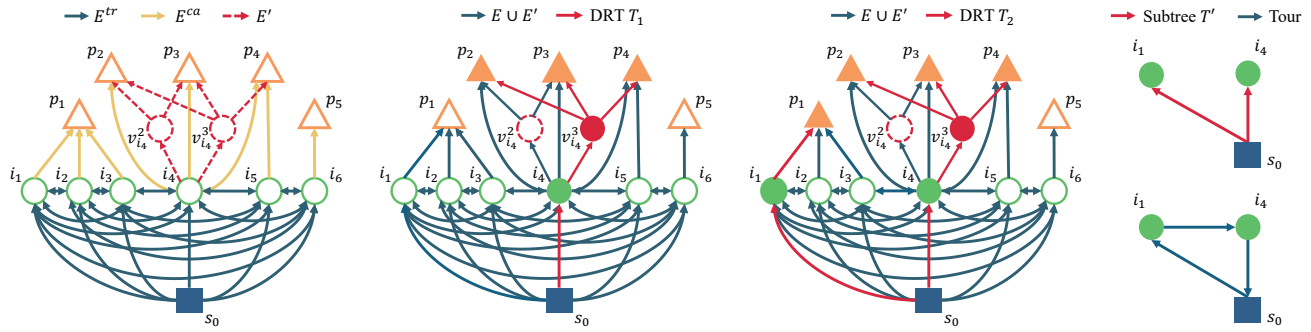
A. Directed Auxiliary Graph Construction (DAGC) Phase

The DAG A is constructed from the PoI network $G = (V, E)$. All V and E in G are retained in A, while edges in $E = (E^{tr} \cup E^{ca})$ are converted into directed edges with associated costs. Specifically, each undirected travel edge $(i,j) \in E^{tr}$ becomes two directed edges with cost t^{tr}_{ij}. Each edge traveling from the depot s_0 to a capture point i corresponds to a directed edge (s_0, i) with cost $t^{tr}_{s_0 i}$. Each capture edge $(p,i) \in E^{ca}$ becomes a directed edge (i,p) with capture cost t^{ca}_{pi}. All other edges not explicitly mentioned are assigned the cost of $+\infty$. Subsequently, the DAGC examines each capture point that captures PoIs shared with other capture points. For each such point, the DAGC sorts its capturing PoIs in a non-decreasing order of their capture time costs. Let n_i denote the number of PoIs capturable at i. If $n_i > 1$, these PoIs are ordered in non-decreasing capture times t^{ca}_{pi}. Let p^k denote the k-th PoI in this sequence, and define $V^k_i = \{p^1, p^2, \ldots, p^k\}$ as the set of first k PoIs, where $k \in \{2, 3, \ldots, n_i\}$. For each V^k_i, a virtual node v^k_i is added, along with a directed edge (i, v^k_i) with cost $t^{ca}_{p^k i}$. Additionally, for every $p \in V^k_i$, a zero-cost edge (v^k_i, p) is added. Let V' denote the set of all virtual nodes, and E' be the set of corresponding virtual bi-directed edges. The final DAG is then defined as $A = (V \cup V',\ E \cup E')$.

Following Fig. 1, Fig. 2(a) illustrates an example for the DAGC phase. Assume that G contains a depot, five PoIs, and six capture points. To construct the DAG A, we first duplicate G and add directionality to E, as depicted in Fig. 2(a). For clarity, edges with infinite cost are omitted. Subsequently, we identify capture point i_4 as the unique point capturing multiple PoIs, where $n_{i_4} = 3$. We assume that the capture time is ordered as $t^{ca}_{p_2 i_4} \leq t^{ca}_{p_3 i_4} \leq t^{ca}_{p_4 i_4}$. This ordering leads to the sets $V^2_{i_4} = \{p_2, p_3\}$ and $V^3_{i_4} = \{p_2, p_3, p_4\}$. For these two sets, we add corresponding virtual nodes $v^2_{i_4}$ and $v^3_{i_4}$. Virtual edges are then created. Zero-cost edges extend from $v^2_{i_4}$ to p_2, p_3 and from $v^3_{i_4}$ to p_2, p_3, p_4. Additional virtual edges are directed from i_4 to $v^2_{i_4}$ with cost $t^{ca}_{p_3 i_4}$ and to $v^3_{i_4}$ with cost $t^{ca}_{p_4 i_4}$. Finally, A includes two virtual nodes and fourteen virtual edges.

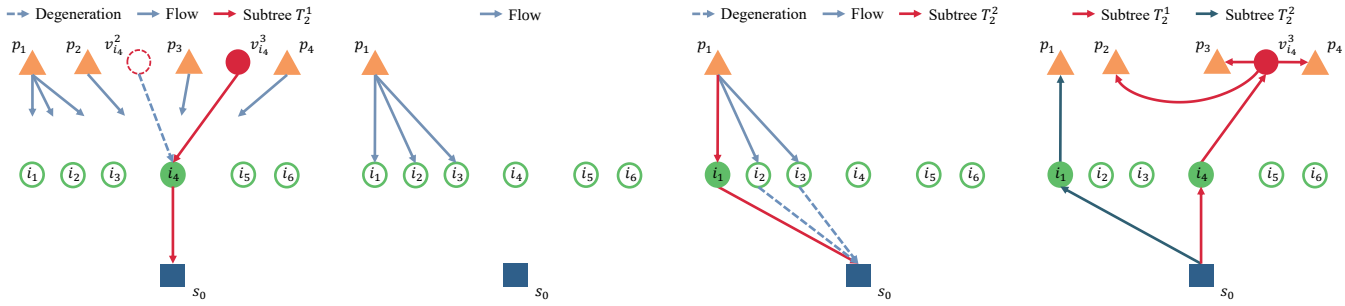
B. PoI and Its Capture Point Selection (PCPS) Phase

After the DAG A is constructed, the PCPS determines a subset of capture points $\hat{S}$ to maximize the total reward $R(\hat{S})$ within the time budget B. For clarity, $R(\hat{S})$ denotes the reward of the set of PoIs covered by the chosen capture points $\hat{S}$. To this end, the PCPS first introduces the marginal gain, defined as $\sigma(\mathcal{S}, \{i\}) = R(\mathcal{S} \cup \{i\}) - R(\mathcal{S})$, where $\mathcal{S} \subseteq S$. Starting from $\hat{S} = \{\emptyset\}$, at each iteration ℓ, the PCPS selects a capture point i with the highest marginal gain $\sigma(\hat{S}, \{i\})$ and adds i into $\hat{S}$. Then, based on A, it constructs a DRT T_ℓ, rooted at the depot, that spans all the captured PoIs by the current $\hat{S}$. The iteration repeats until the cost of T_ℓ exceeds $\frac{B}{2}$ and returns the final DRT $T = T_{\ell-1}$. Finally, the PCPS selects a subset of capture points $\hat{S}$ and their PoIs based on the results of T.



(a) DAG A (b) DRT T_1 (c) DRT T_2 (d) Final tour

Fig. 2. Example of TURA.



(a) Round = 1, elapsed time = 1 (b) Round = 2, elapsed time = 2 (c) Round = 2, elapsed time = 3 (d) $T_2 = (T_2^1 \cup T_2^2)$

Fig. 3. Example of DRT construction.

Following Fig. 2(a), Fig. 2(b) and 2(c) illustrate an example for the PCPS phase. For clarity, edges with infinite cost are omitted. Assume the time budget $B = 12$, and that the rewards for each PoI are $(r_{p_1}, r_{p_2}, r_{p_3}, r_{p_4}, r_{p_5}) = (3, 6, 1, 5, 2)$. The reward function follows $R(\mathcal{P}) = \sum_{p \in \mathcal{P}} r_p$, where $\mathcal{P} \subseteq P$. In addition, each edge from capture point $i \in S_p$ to its corresponding PoI p has the cost $t^{ca}_{pi} = 2$, and each edge from the depot to capture point and between any two capture points has the cost $t^{tr}_{s_0 i} = 1$ and $t^{tr}_{ij} = 1$. In the first iteration, $v^3_{i_4}$ (i.e., i_4) is selected with the highest marginal gain $\sigma = 12$, and T_1 is constructed. The second iteration selects i_1 with the highest $\sigma = 15 - 12 = 3$, resulting in T_2. Since the cost of T_2 is 6, the PCPS ends with T_2, deriving the final DRT T.

The DRT T_ℓ construction is detailed as follows. The idea is to let each captured PoI be *a subtree's leaf*. Each subtree starting from the leaf grows simultaneously at first, covers the edge with its cost, and combines together if multiple subtrees encounter at a node. If the subtree combination occurs, the growth rate will accelerate as the subtree can span more PoIs. Specifically, we consider *a flow* starting from each captured PoI. The flow is increasing according to its flow rate (i.e., flow unit per time unit), which is set to the number of PoIs spanned by the flow. The edge is *saturated* until the flow reaches its cost (i.e., the edge fully covered by the flow).[3] The round γ terminates until one of flows reaches the root (i.e., depot) and returns a subtree T^γ_ℓ spanning a subset of captured PoIs. Afterward, we remove the spanned PoIs and repeat the above procedures until all the captured PoIs are spanned. Finally, we have $T_\ell = \cup_\gamma T^\gamma_\ell$, and the construction of DRT T_ℓ is finished.

Fig. 3 illustrates the details for the DRT T_2 construction. In the first round, as shown in Fig. 3(a), the flow starts from each captured PoI. When the elapsed time is one unit, a flow unit is from p_1, p_2, p_3, p_4, and two and three flow units are from $v^2_{i_4}$ and $v^3_{i_4}$, respectively. This is because $v^2_{i_4}$ and $v^3_{i_4}$ connect two and three captured PoIs (i.e., their flow rates are two and three). Other flow rates are only one. Recall that the edge cost $t^{ca}_{pi} = 2$ and $t^{tr}_{ij} = 1$. The flow from $v^3_{i_4}$ reaches the root s_0. The first round terminates, returns the subtree T^1_2, and removes the spanned p_2, p_3, p_4 and the edge $(v^2_{i_4}, i_4)$ due to the degeneration. In the second round, the above procedure repeats. When the elapsed time is two units, as shown in Fig. 3(b), the edges (p_1, i_1), (p_1, i_2), (p_1, i_3) are saturated. When the elapsed time is three units, as shown in Fig. 3(c), the flow from p_1 reaches s_0, resulting in the construction of another subtree T^2_2. Due to the degeneration, two edges are removed. Finally, $T_2 = (T^1_2 \cup T^2_2)$ is constructed as shown in Fig. 3(d).

C. UAV Tour Generation (UTG) Phase

The selection of PoI and its capture point is completed via the marginal gain and the DRT construction. To generate a UAV tour that visits these selected capture points, the UTG prunes T into a subtree T' by retaining only the depot, the selected capture points, and the travel edges connecting them. Finally, the UTG performs a DFS traversal on the subtree T'. It starts at the depot, explores as far as possible along each

[3]If a PoI reaches a node with two or more distinct paths of saturated edges or with a path containing a cycle, we say the flow is *degenerate*. For each time unit when there is an edge saturated, we examine the flow degeneration. If the degeneration exists, we will remove the last saturated *edge*. In the case of a tie, we prioritize selecting the edge with its node spanning most PoIs.

branch, backtracks only when getting stuck, and returns to the depot at the end. Due to the triangle inequality, the UTG generates a UAV tour by shortcutting. In this way, the tour contains each selected capture point exactly once. As shown in Fig. 2(d), we prune T to obtain T', which has a root s_0 and two leaves i_1, i_4. The DFS traversal is $s_0 \to i_1 \to s_0 \to i_4 \to s_0$. By shortcutting, the final tour will be $s_0 \to i_1 \to i_4 \to s_0$.

The TURA is an approximation algorithm. Due to the page limit, the proofs are detailed in Appendices B and C of [13].

Theorem 2. The TURA is an $O(\frac{1}{\epsilon})$-approximation algorithm, where $\epsilon = \frac{L_{min}}{L_{max}}$. Here, $L_{\min}$ denotes the minimum shortest-path time from the depot to a capture point, and L_{max} is the maximum shortest-path time from the depot to a PoI.

V. Performance Evaluation

A. Simulation Settings

1) Dataset: To make our simulation more realistic, we use two real-world datasets for PoI networks and video capture time. The first dataset is **Carto PoI Data-Canada** [14], which includes the latitude and longitude information of various PoIs in Canada. We extract a sparse and dense network with the area of 1.5×1.5 km^2, respectively, where the sparse network includes 8 main categories of PoIs and the dense network includes 8 main and other 183 categories of PoIs. The UAV brand is Matrice 600 Pro [15] and the camera is Zenmuse z30 [16]. The flight time is calculated by the distance and flight speed, where the UAV maximum speed is 65 km/hr [15]. The camera coverage is 447×447 m^2 with the maximum height of 360 m, calculated by field-of-view (FOV) [16], [17], and thus we set the coverage radius as 200 m centered at a capture point. We then assign the capture points for PoIs according to the coverage radius. The PoI can only be captured in the radius. The second dataset is **DL3DV-10K** [18], which includes $10,510$ videos captured from 65 types of PoIs and their video lengths. We randomly generate the capture time for each PoI based on its distribution of different video lengths.

2) Parameter Settings: In both sparse and dense networks, the area ranges from 0.5×0.5 to 1.5×1.5 km^2, with a default of 1.5×1.5 km^2. In the area of 1.5×1.5 km^2 with the coverage radius of 200 m, there are about 6 and 16 PoIs and 7 and 27 capture points in the sparse and dense networks. The time budget B ranges from 1000 to 2000, with a default of 1500. The default reward for each PoI is set within the range of 1 and 100, and we also randomly assign the reward of each PoI from different ranges, i.e., $[1, 50]$ to $[1, 150]$.

3) Baselines: We compare the performance of the TURA with four baselines. **1) Greedy (GRD)**: At each iteration, it selects a capture point capturing PoIs with the highest ratio of reward to the shortest path (from the depot to the PoI with the longest capture time) and builds a directed tree connecting all the selected capture points, until the time costs exceed $\frac{B}{2}$. Finally, it performs a DFS traversal with shortcutting to generate a tour. **2) Weighted Set Cover (WSC)**: Similar to the first method, the only difference is to select a capture point capturing PoIs with the highest reward-to-capture-time ratio. **3) Group Steiner Tree (GST)** [19]: At each iteration, it selects a capture point with the highest marginal gain and builds a tree spanning all the PoIs of the selected capture points, until the time costs exceed $\frac{B}{2}$. Similarly, it performs a DFS traversal with shortcutting to generate a tour. **4) OPT**: It uses Gurobi to derive the optimal solution of MILP (1a)−(1n). In addition, each simulation result is averaged over 15 trials.

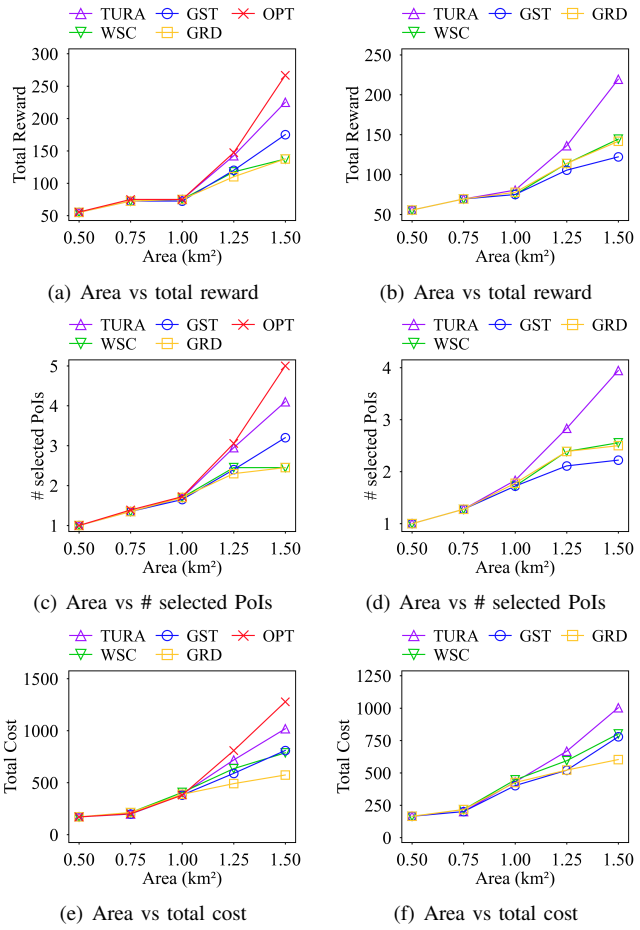
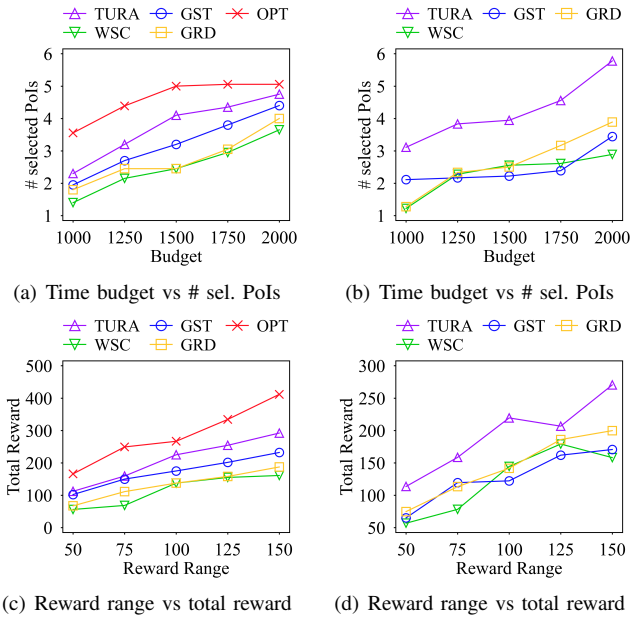


Fig. 4. Effect of area, where (a), (c), (e) are sparse networks and (b), (d), (f) are dense networks.

B. Numerical Results

The numerical results in Figs. 4 and 5 demonstrate the effects of various parameters on the different performance metrics. The proposed TURA jointly selects capture points and PoIs, achieving the highest total reward within the tour and capture time budget, and consistently outperforming other baselines. The results are explained in detail as follows.

1) Effect of Area: The effects of the area on the total reward, number of selected PoIs, and total cost in the sparse and dense networks are shown in Fig. 4. In Fig. 4(a)–4(b), the total reward of all algorithms increases as the area grows, since a larger area contains more valuable PoIs. Compared to the other methods, the proposed TURA can achieve the highest total reward, close to the OPT, even as the area becomes large. In Fig. 4(c)–4(d), the number of selected PoIs also exhibits an increasing trend. This is because, there include more valuable PoIs along with capture points, resulting in more PoIs being selected. Finally, Fig. 4(e)–4(f) show that the TURA utilizes the available time budget efficiently, achieving total costs that closely approach



(a) Time budget vs # sel. PoIs (b) Time budget vs # sel. PoIs

(c) Reward range vs total reward (d) Reward range vs total reward

Fig. 5. Effects of time budget and reward range, where (a), (c) are sparse networks and (b), (d) are dense networks.

that of the OPT. As a result, all comparative methods yield lower total rewards. Specifically, the GRD overlooks high-reward PoIs, the WSC fails to account for both high-reward PoIs and tour cost, and the GST neglects capture cost. The latter two also lead to relatively higher total costs. In contrast, the TURA effectively selects more valuable PoIs and achieves a higher total reward, successfully addressing the first challenge.

2) Effect of Time Budget: The effect of the time budget from 1000 to 2000 on the number of selected PoIs is shown in Fig. 5(a) and 5(b) for the sparse and dense networks, respectively. As expected, a larger time budget allows all algorithms to select more PoIs, as the UAV has greater flexibility to travel and perform captures. Notably, the proposed TURA makes more effective use of the additional available budget by leveraging marginal gain-based capture point selection and constructing a DRT to span the selected PoIs. This strategy allows it to prioritize high-value PoIs and minimize redundant tour and capture time costs, solving the second and third challenges.

3) Effect of Reward Range: The effect of the reward range from $[1, 50]$ to $[1, 150]$ on the total reward is shown in Fig. 5(c) and 5(d) for the sparse and dense networks, respectively. As the reward range increases, the variation in reward values among PoIs becomes larger, leading to an overall increase in the total reward achieved by all algorithms. In this context, the proposed TURA consistently outperforms the other methods, achieving a higher total reward. This trend is more evident in the dense network, where the availability of more capture points allows the algorithm to better exploit high-reward PoIs.

4) Comparison of Running Time: Table I shows the average running time performance on the different areas with the baselines. The average running time in all methods increases with the areas. Although the OPT can obtain the optimal solution, it requires much longer than the other methods. In contrast, the proposed TURA can achieve the solution close to the OPT in a short running time within 0.035 seconds.

TABLE I
RUNNING TIME OF DIFFERENT BASELINES AVG. OVER 15 TRIALS (SEC)

Area (km^2)	0.5×0.5	0.75×0.75	1×1	1.25×1.25	1.5×1.5
TURA	2.6×10^{-5}	5.7×10^{-5}	0.002	0.007	0.035
GRD	2.2×10^{-6}	2.6×10^{-6}	4.1×10^{-6}	6.7×10^{-6}	1.0×10^{-5}
WSC	2.2×10^{-6}	2.5×10^{-6}	3.8×10^{-6}	6.1×10^{-6}	8.0×10^{-6}
GST	0.028	0.038	0.057	0.096	0.187
OPT	0.027	0.025	0.034	0.774	333.656

VI. CONCLUSION

This paper addresses an optimization problem called TURP, which aims to maximize the total reward from capturing PoIs within the time budget. To tackle this problem, we propose an approximation algorithm named TURA with three phases. In the first phase, a DAG is constructed to model joint PoI capture using virtual nodes and edges. In the second phase, a subset of capture points is iteratively selected based on the marginal gain and a DRT is built to span all the selected PoIs. In the last phase, a UAV tour is generated by pruning the tree and applying a DFS traversal with shortcutting. Overall, the paper proves the NP-hardness and approximation ratio in the theoretical analysis, and simulation results demonstrate that the proposed TURA outperforms other baselines by up to 67%.

REFERENCES

[1] S.-W. Chang, J.-J. Kuo *et al.*, "Near-optimal UAV deployment for delay-bounded data collection in IoT networks," in *IEEE INFOCOM*, 2024.

[2] J. Zhang *et al.*, "Ordered submodularity-based value maximization of UAV data collection in earthquake areas," *IEEE Trans. Netw. Sci. Eng.*, vol. 11, no. 5, pp. 4886–4897, 2024.

[3] N. Van Cuong, Y.-W. P. Hong, and J.-P. Sheu, "UAV-enabled image capture and wireless delivery for on-demand surveillance tasks," *IEEE Trans. Wirel. Commun.*, vol. 23, no. 10, pp. 12 995–13 010, 2024.

[4] Y. Liang *et al.*, "Nonredundant information collection in rescue applications via an energy-constrained UAV," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2945–2958, 2019.

[5] W. Xu *et al.*, "Reward maximization for disaster zone monitoring with heterogeneous UAVs," *IEEE/ACM Trans. Netw.*, vol. 32, no. 1, pp. 890–903, 2024.

[6] X. Gao, X. Zhu, and L. Zhai, "AoI-sensitive data collection in multi-UAV-assisted wireless sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 22, no. 8, pp. 5185–5197, 2023.

[7] W. Liang *et al.*, "Approximation algorithms for charging reward maximization in rechargeable sensor networks via a mobile charger," *IEEE/ACM Trans. Netw.*, vol. 25, no. 5, pp. 3161–3174, 2017.

[8] Y. Ma, W. Liang, and W. Xu, "Charging utility maximization in wireless rechargeable sensor networks by charging multiple sensors simultaneously," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1591–1604, 2018.

[9] T.-W. Kuo, K. C.-J. Lin, and M.-J. Tsai, "Maximizing submodular set function with connectivity constraint: Theory and application to networks," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 533–546, 2015.

[10] G. D'Angelo, E. Delfaraz, and H. Gilbert, "Budgeted out-tree maximization with submodular prizes," in *ISAAC*, 2022.

[11] X. Fang, H. Gao, J. Li, and Y. Li, "Application-aware data collection in wireless sensor networks," in *IEEE INFOCOM*, 2013.

[12] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, pp. 265–294, 1978.

[13] C.-A. Yang *et al.*, "Dual time-constrained UAV routing for reward maximization under joint multi-PoI capture (appendix)," Oct 2025. [Online]. Available: https://reurl.cc/ORN269

[14] "Carto PoI Data," 2025. [Online]. Available: https://reurl.cc/Gp3ggD

[15] "Matrice 600 Pro," 2025. [Online]. Available: https://reurl.cc/OryQ5X

[16] "DJI Zenmuse z30," 2025. [Online]. Available: https://reurl.cc/WN2z8x

[17] L. H. Nam, L. Huang, X. J. Li, and J. F. Xu, "An approach for coverage path planning for UAVs," in *IEEE AMC*, 2016.

[18] "DL3DV-10K dataset," 2025. [Online]. Available: https://reurl.cc/nv8xAv

[19] N. Garg, G. Konjevod, and R. Ravi, "A polylogarithmic approximation algorithm for the group Steiner tree problem," in *SODA*, 1998.

APPENDIX A
PROOF OF THEOREM 1

We prove the theorem by reducing the maximum submodular set function (MS) problem [1] to the TURP. Given a non-decreasing submodular set function f on the set of subsets of $\mathcal{V}$ and a positive integer k, the MS problem asks for a subset $\mathcal{S} \subseteq \mathcal{V}$ such that 1) $|\mathcal{S}| \leq k$ and 2) $f(\mathcal{S})$ is maximized. To accomplish the reduction from the MS problem to the TURP, we have to show how to construct an instance of the TURP (i.e., $V = \{P \cup S\}, E = \{E^{tr} \cup E^{ca}\}, t_{ij}^{tr}, t_{pi}^{ca}, B, R$) for the MS problem. The construction steps are detailed as follows.

First, we consider there is no traveling cost (i.e., $t_{ij}^{tr} = 0$). For each element in $\mathcal{V}$, we create a corresponding PoI p and add it to P. Then, we create a corresponding capture point i covering each PoI and add each edge (p, i) to E^{ca} and capture point i to S_p. The capturing cost of each edge t_{pi}^{ca} is set to 1. For a positive integer k, we create a corresponding budget B. For a submodular set function f, we create a corresponding reward function R. The above reduction can be done in the polynomial time. In this way, the MS problem can be solved if the solution of the TURP can be found. However, the MS problem is a known NP-hard problem and can only be solved when P = NP. Therefore, the theorem holds.

APPENDIX B
LEMMA 1

In the second phase PCPS, we iteratively select a capture point with the highest marginal gain until the cost exceeds $\frac{B}{2}$. Let k denote the number of capture points selected by this greedy algorithm. For finding a subset $\hat{S} \subseteq S \setminus \{s_0\}$ such that $|\hat{S}| \leq k$ and $R(\hat{S})$ is maximized, we show that this greedy algorithm can achieve an approximation ratio as follows.

Lemma 1. The greedy algorithm for capture point selection is an $\frac{e}{e-1}$-approximation algorithm.

Proof. Let $\hat{S}_\ell$ denote a subset of selected capture points after ℓ iterations, where $\hat{S}_0 = \{\emptyset\}$ and $\hat{S} = \hat{S}_k$. At each iteration ℓ, the algorithm iteratively selects a capture point s_ℓ with the highest marginal gain $\sigma(\hat{S}_{\ell-1}, \{s_\ell\}) = R(\hat{S}_{\ell-1} \cup \{s_\ell\}) - R(\hat{S}_{\ell-1})$. Then, we define $\delta_\ell = R(S^*) - R(\hat{S}_{\ell-1})$, where S^* represents the optimal capture point selection such that $|S^*| \leq k$ and $R(S^*)$ is maximized. By submodularity, we have

$$R(S^*) - R(\hat{S}_{\ell-1}) \leq \sum_{i \in S^* \setminus \hat{S}_{\ell-1}} \sigma(\hat{S}_{\ell-1}, \{i\}). \tag{2}$$

Then, since the algorithm always selects a capture point with the highest marginal gain, we have the following equation:

$$\sigma(\hat{S}_{\ell-1}, \{i\}) \leq \sigma(\hat{S}_{\ell-1}, \{s_\ell\}),\ \forall i \in S^* \setminus \hat{S}_{\ell-1}. \tag{3}$$

By (2) and (3), it follows that

$$\begin{aligned} \sum_{i \in S^* \setminus \hat{S}_{\ell-1}} \sigma(\hat{S}_{\ell-1}, \{i\}) &\leq \sum_{i \in S^* \setminus \hat{S}_{\ell-1}} \sigma(\hat{S}_{\ell-1}, \{s_\ell\}) \\ &\leq |S^* \setminus \hat{S}_{\ell-1}| \cdot \sigma(\hat{S}_{\ell-1}, \{s_\ell\}) \\ &\leq k \cdot \sigma(\hat{S}_{\ell-1}, \{s_\ell\}) \end{aligned} \tag{4}$$

and

$$R(S^*) - R(\hat{S}_{\ell-1}) \leq k \cdot \sigma(\hat{S}_{\ell-1}, \{s_\ell\}). \tag{5}$$

By rearranging (5), we can obtain

$$\sigma(\hat{S}_{\ell-1}, \{s_\ell\}) \geq \frac{R(S^*) - R(\hat{S}_{\ell-1})}{k} = \frac{\delta_\ell}{k}. \tag{6}$$

Therefore,

$$\delta_{\ell+1} = \delta_\ell - \sigma(\hat{S}_{\ell-1}, \{s_\ell\}) \leq \delta_\ell \left(1 - \frac{1}{k}\right). \tag{7}$$

Recursively, $\delta_{\ell+1} \leq \delta_1 \left(1 - \frac{1}{k}\right)^\ell$ and $\delta_1 = R(S^*)$. The algorithm selects k capture points, and thus $\delta_{k+1} \leq \delta_1 \left(1 - \frac{1}{k}\right)^k$. Since $\lim_{k \to \infty} \left(1 - \frac{1}{k}\right)^k \leq e^{-1}$, we have

$$R(S^*) - R(\hat{S}_k) \leq R(S^*) \cdot \left(1 - \frac{1}{k}\right)^k \leq R(S^*) \cdot e^{-1}. \tag{8}$$

Finally, since $\hat{S} = \hat{S}_k$,

$$R(S^*) - R(\hat{S}) \leq R(S^*) \cdot e^{-1}. \tag{9}$$

By rearranging, we have the total reward at most:

$$R(S^*) \leq \frac{e}{e-1} R(\hat{S}). \tag{10}$$

Therefore, the lemma holds. □

APPENDIX C
PROOF OF THEOREM 2

In this proof, we analyze the gap between the TURA and the optimal solution. Before the proof, we first define the tree version of the TURP. The only difference is that the UAV route forms a tree spanning the selected capture points rooted at the depot, rather than a tour that visits each point and returns to the depot. Let S_{tour}^* and S_{tree}^* denote the optimal solution with the time budget of B under the tour and tree version, respectively. Let L_{max} denote the maximum shortest path (including travel and capture time) from the depot to a PoI. The total number of selected capture points in the PCPS is at least $\frac{B}{2L_{max}}$. Let $\hat{S}$ denote a subset of capture points selected by the TURA and assume $|\hat{S}| = \frac{B}{2L_{max}}$. The proof is detailed as follows.

Let $S^*_{\frac{B}{2L_{max}}}$ denote the optimal solution for the problem that selects a subset of capture points such that $|S^*_{\frac{B}{2L_{max}}}| \leq \frac{B}{2L_{max}}$ and $R(S^*_{\frac{B}{2L_{max}}})$ is maximized. Based on Lemma 1, we have

$$R(S^*_{\frac{B}{2L_{max}}}) \leq \frac{e}{e-1} \cdot R(\hat{S}). \tag{11}$$

Then, we divide the optimal tree that spans a subset of capture points S_{tree}^* into multiple connected subtrees, each spanning a subset of capture points S^t and $|S^t| \leq \frac{B}{2L_{max}}$. Thus, we have

$$R(S^t) \leq R(S^*_{\frac{B}{2L_{max}}}) \tag{12}$$

since $S^*_{\frac{B}{2L_{max}}}$ is the optimal solution that maximizes the total reward under the constraint $|S^*_{\frac{B}{2L_{max}}}| \leq \frac{B}{2L_{max}}$. Let L_{min} denote the minimum shortest path (including travel and capture time) from the depot to a capture point. We can obtain at most

$\frac{B}{L_{min}} \cdot \frac{2L_{max}}{B} = \frac{2L_{max}}{L_{min}}$ subtrees from the depot to at most $\frac{B}{2L_{max}}$ capture points. That is,

$$R(S^*_{tree}) \leq R(\cup_{t=1}^{\frac{2L_{max}}{L_{min}}} S^t) = \sum_{t=1}^{\frac{2L_{max}}{L_{min}}} R(S^t). \tag{13}$$

The second equality is because the spanned capture points are the same. After that, under the time budget of B, we have

$$R(S^*_{tour}) \leq R(S^*_{tree}). \tag{14}$$

Since a tour can be transformed into a tree by removing any one of its edges, the tour with a subset of capture points S^*_{tour} removing an arbitrary edge can become a tree with the same reward. The optimal tree with S^*_{tree} achieves a higher reward than the tree from the tour with S^*_{tour}. Thus, the inequality (14) holds. Finally, we have the total reward at most:

$$\begin{aligned} R(S^*_{tour}) \leq R(S^*_{tree}) &\leq \sum_{t=1}^{\frac{2L_{max}}{L_{min}}} R(S^t) \leq \sum_{t=1}^{\frac{2L_{max}}{L_{min}}} R(S^*_{\frac{B}{2L_{max}}}) \\ &\leq \sum_{t=1}^{\frac{2L_{max}}{L_{min}}} \frac{e}{e-1} \cdot R(\hat{S}) \\ &\leq \frac{2L_{max}}{L_{min}} \cdot \frac{e}{e-1} \cdot R(\hat{S}) \\ &\leq O(\frac{L_{max}}{L_{min}}) \cdot R(\hat{S}) \\ &= O(\frac{1}{\epsilon}) \cdot R(\hat{S}). \end{aligned} \tag{15}$$

Therefore, the approximation ratio is $O(\frac{1}{\epsilon})$, where $\epsilon = \frac{L_{min}}{L_{max}}$ with $\epsilon \in (0, 1)$. Note that the time budget will not be violated. Since the UTG performs a DFS traversal on the pruned tree to generate the tour, every edge in the tree will be visited twice. The tour cost is at most twice of the tree due to triangle inequality. Thus, the time cost of the TURA is at most $\frac{B}{2} \cdot 2 = B$. In the following, we analyze the time complexity.

Time Complexity. The second PCPS phase dominates the overall time complexity. Specifically, the first phase takes $O(|S||P|)$ time to construct a DAG. The second phase takes $O(|S|^2 \cdot |P| \cdot |P \cup V'| \cdot |E \cup E'|)$ time to select a capture point with the highest marginal gain and construct a DRT. The third phase takes $O(|S| + |E^{tr}|)$ time to perform a DFS traversal and shortcutting. Therefore, the TURA's time complexity is $O(|S|^2 \cdot |P| \cdot |P \cup V'| \cdot |E \cup E'|)$ in polynomial time. ■

REFERENCES

[1] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions—I," *Math. Program.*, vol. 14, pp. 265–294, 1978.