Approximation Algorithms for the Generalized Team Orienteering Problem and Its Applications

Wenzheng Xu, *Member, IEEE*, Weifa Liang, *Senior Member, IEEE*, Zichuan Xu, *Member, IEEE*, Jian Peng, Dezhong Peng, *Member, IEEE*, Tang Liu, Xiaohua Jia, *Fellow, IEEE*, and Sajal K. Das, *Fellow, IEEE*

***Abstract*— In this article we study a generalized team orienteering problem (GTOP), which is to find service paths for multiple homogeneous vehicles in a network such that the profit sum of serving the nodes in the paths is maximized, subject to the cost budget of each vehicle. This problem has many potential applications in IoTs and smart cities, such as dispatching energy-constrained mobile chargers to charge as many energy-critical sensors as possible to prolong the network lifetime. In this article, we first formulate the GTOP problem, where each node can be served by different vehicles, and the profit of serving the node is a submodular function of the number of vehicles serving it. We then propose a novel $(1-(1/e)^{\frac{1}{2+\epsilon}})$-approximation algorithm for the problem, where ϵ is a given constant with $0 < \epsilon \leq 1$ and e is the base of the natural logarithm. In particular, the approximation ratio is about 0.33 when $\epsilon = 0.5$. In addition, we devise an improved approximation algorithm for a special case of the problem where the profit is the same by serving a node once and multiple times. We finally evaluate the proposed algorithms with simulation experiments, and the results of which are very promising. Especially, the profit sums delivered by the proposed algorithms are up to 14% higher than those by existing algorithms, and about 93.6% of the optimal solutions.**

***Index Terms*— Multiple vehicle scheduling, the generalized team orienteering problem, approximation algorithms, submodular function.**

Manuscript received January 22, 2020; revised June 25, 2020, August 6, 2020, and September 24, 2020; accepted September 25, 2020; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor R. Lo Cigno. Date of publication October 7, 2020; date of current version February 17, 2021. The work of Wenzheng Xu was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61602330 and in part by the Sichuan Science and Technology Program under Grant 2018GZDZX0010 and Grant 2017GZDZX0003. The work of Weifa Liang was supported by the Australian Research Council Discovery Project under Grant DP200101985. The work of Jian Peng was supported in part by the National Key Research and Development Program of China under Grant 2017YFB0202403 and in part by the Sichuan Key Research and Development Program under Grant 2018GZ0098. The work of Dezhong Peng was supported by NSFC under Grant 61971296 and Grant U19A2078. The work of Tang Liu was supported in part by NSFC under Grant 62072320 and in part by the Scientific Research Fund of Sichuan Provincial Education Department under Grant 18ZA0404. The work of Sajal K. Das was supported in part by NSF under Grant CCF-1725755, Grant CNS-1818942, and Grant CNS-1545050. *(Corresponding author: Tang Liu.)*

Wenzheng Xu and Jian Peng are with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: wenzheng.xu3@gmail.com; jianpeng@scu.edu.cn).

Weifa Liang is with the Research School of Computer Science, The Australian National University, Canberra, ACT 2601, Australia (e-mail: wliang@cs.anu.edu.au).

Zichuan Xu is with the School of Software, Dalian University of Technology, Dalian 116024, China (e-mail: z.xu@dlut.edu.cn).

Dezhong Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China, and also with the Peng Cheng Laboratory, Shenzhen 518052, China (e-mail: pengdz@scu.edu.cn).

Tang Liu is with the College of Computer Science, Sichuan Normal University, Chengdu 610068, China (e-mail: liutang@sicnu.edu.cn).

Xiaohua Jia is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: csjia@cityu.edu.hk).

Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA (e-mail: sdas@mst.edu).

Digital Object Identifier 10.1109/TNET.2020.3027434

I. INTRODUCTION

IN THIS article we consider a *generalized team orienteering problem (GTOP)*, which has wide applications in the domains such as Internet of Things (IoTs) and smart cities [18], [30], [32]. We here briefly introduce its two potential applications: **(i)** Dispatching multiple mobile chargers to recharge sensors in rechargeable sensor networks; and **(ii)** scheduling Unmanned Aerial Vehicles (UAVs) to monitor disaster zones.

We start with the first application of the problem, that is to dispatch multiple mobile chargers to recharge sensors. On one hand, sensors usually are powered by energy-limited batteries. On the other hand, they consume their battery energy when they perform sensing, transmit and receive sensing data. They will run out of energy eventually. An effective solution to this sensor energy expiration problem is to dispatch the mobile chargers to recharge energy-critical sensors, where a mobile charger can move to the location of an energy-critical sensor, and replenish its energy to the sensor via wireless energy transfer [10], [11], [22], [24]–[26], [31], [34]–[38], [41]. Fig. 1 illustrates the employment of two mobile chargers to recharge sensors in a wireless rechargeable sensor network. When many energy-critical sensors need to be charged, the two mobile chargers may not be enough to charge all the sensors to their full energy capacities, due to the limited energy capacity on each of the two chargers. Therefore, a fundamental problem is to schedule the two chargers to recharge a portion of energy-critical sensors such that the profit sum of the charged sensors is maximized, subject to the energy capacity on each mobile charger. It is understood that the more profit can be obtained by charging a sensor with low residual energy than that by charging a sensor with high residual energy [22], [24].

We then introduce the second application of the GTOP problem by scheduling multiple Unmanned Aerial Vehicles (UAVs) to monitor disaster zones. Lightweight UAVs, such as a DJI phantom 4 Pro, are widely used in aerial photography, precision agriculture, disaster rescue, etc [14], [20], [23], [27].

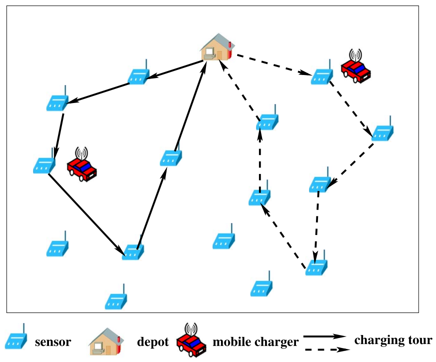


Fig. 1. An example of the generalized team orienteering problem for dispatching two mobile chargers to recharge energy-critical sensors in a sensor network.

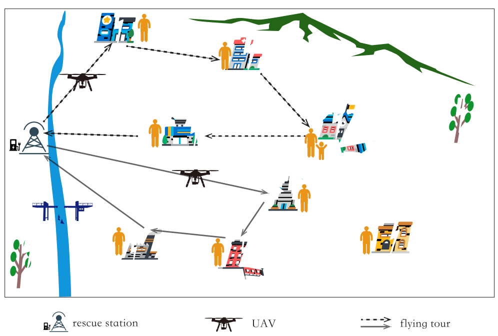


Fig. 2. An example of the GTOP problem in scheduling two UAVs to monitor PoIs in a disaster zone.

For example, when a disaster (such as an earthquake or a flooding) occurs, the most critical issue is to rescue people in danger as quickly as possible. However, transportation and communication infrastructures may have been destroyed in the disaster already. In this scenario, UAVs can be used to aid disaster rescuing, by dispatching the UAVs to take photos or videos for Points of Interest (PoIs), e.g., malls, schools, office buildings, in the disaster area, and transmitting these invaluable information (i.e., photos or videos) to a nearby rescue station for rescue decision making, see Fig. 2. For example, UAVs were used for taking photos in order to discover people in danger after Hurricane "Irma" struck Florida in 2017 [13]. In this application, it is desirable to collaboratively monitor as many PoIs as possible, since the maximum flying duration of each UAV is limited due to its limited energy capacity, and the maximum flying duration of a 'DJI phantom 4 Pro' UAV usually is only around 30 minutes.

Other important applications of the GTOP problem include scheduling multiple repairmen to repair sharing bikes, e.g., Mobikes, located around a city [30], and dispatching a fleet of autonomous vehicles to deliver goods to different households in the scenario of smart cities.

The GTOP problem is a generalization of the traditional orienteering problem [18], [32] that is defined as follows. Given a complete graph $G = (V, E)$ and a cost budget B, let each edge $(v_i, v_j) \in E$ be associated with a cost $c(v_i, v_j)$. Assume that the edge costs in G satisfy the triangle inequality. Also, let there be a profit $u(v_i)$ of serving a node $v_i \in V$. The *orienteering problem* is to find a simple path P in G from a source node s to a destination node t, such that the profit sum of the nodes in path P, denoted as $\sum_{v_i \in P} u(v_i)$, is maximized, subject to the constraint that the total cost of the edges in P is no greater than the cost budget B.

Due to the wide applications, the orienteering problem has been extensively studied in the literature. Chekuri *et al.* [8] recently proposed a $\frac{1}{2+\epsilon}$-approximation algorithm for the problem, when the starting node s and ending node t of a tour may be different, where ϵ is a given constant with $0 < \epsilon \le 1$. Moreover, Paul *et al.* [30] devised a $\frac{1}{2}$-approximation algorithm for the problem when the nodes s and t are co-located (i.e., $s = t$). Although the orienteering problem has been extensively studied, many applications need to find paths for multiple vehicles rather than just one vehicle. Here, the meaning of a vehicle is broad; it may be a mobile charger or a UAV, depending on the application scenario. For example, in a large-scale sensor network, we may schedule multiple mobile chargers to recharge as many energy-critical sensors as possible. To the best of our knowledge, there are no performance-guaranteed algorithms for such multi-vehicle case.

In this article, we study the GTOP problem, which is to find service paths for $K > 1$ homogeneous vehicles where each path starts at node s and ends at node t, such that the profit sum of serving the nodes in the K paths is maximized, subject to the cost budget on each vehicle. Furthermore, we consider two scenarios of the problem in practical applications, such as:

(i) Node costs may also be considered in addition to edge costs. In this case, the cost of a path becomes the sum of its edge and node costs. For example, in a sensor network, a mobile charger consumes its energy on both mechanical movements (i.e., edge costs) and recharging sensors (i.e., node costs).

(ii) Each node may be served by multiple vehicles (rather than by only one vehicle) and a nondecreasing submodular function can be adopted to model the profit obtained by serving a node. In other words, the more vehicles serve a node, the less marginal profit it will collect from the node. For example, consider the deployment of multiple UAVs to monitor PoIs in a disaster area, where two or more UAVs can take photos for the same PoI, thereby obtaining more accurate information about the PoI [24]. However, photos taken by different UAVs may contain redundant information. In this scenario, it is appropriate to adopt a submodular function to model the nonredundant information of the photos taken by different UAVs.

The novelty of this article lies in formulating a novel problem, namely, the *generalized team orienteering problem* that has many potential applications in the context of IoTs and smart cities, and developing the very first approximation algorithms with provable approximation ratios for the problem.

Our major contributions are summarized as follows. (1) We are the first to study the GTOP problem where each node can be served by multiple vehicles and the profit collected by serving a node is a submodular function of the number of vehicles serving the node. (2) We propose a novel $(1-(1/e)^{\frac{1}{2+\epsilon}})$-approximation algorithm for the problem, where ϵ is a given constant with $0 < \epsilon \le 1$ and e is the base of the natural logarithm. In particular, the approximation ratio is about 0.33 when

$\epsilon = 0.5$. (3) We devise an improved approximation algorithm for a special case of the problem where the profit collected from a node is the same no matter how many times the node has been served by one or multiple vehicles. (4) We evaluate the proposed algorithms against benchmarks with simulations. Experiment results show that the profit sums delivered by the proposed algorithms are up to 14% higher than those by existing algorithms and 93.6% of the optimal solutions.

The rest of the paper is organized as follows. Section II reviews related works. Section III introduces preliminary concepts. Sections IV and V respectively propose approximation algorithms for the GTOP problem and its special case. Section VI evaluates the proposed algorithms empirically, and Section VII concludes the paper.

II. Related Work

The orienteering problem and its variants have attracted a lot of attentions due to their wide applications [18], [32]. For the orienteering problem in metric graphs, Blum *et al.* [4] proposed the first constant approximation algorithm with a ratio of $\frac{1}{4}$. Bansal *et al.* [2] shortly improved the approximation ratio to $\frac{1}{3}$. Chekuri *et al.* [8] recently further improved the ratio to $\frac{1}{2+\epsilon}$ when the starting node s and the ending node t may be different, where ϵ is a given constant with $0 < \epsilon \leq 1$, while Paul *et al.* [30] devised a $\frac{1}{2}$-approximation algorithm when nodes s and t are co-located, i.e., $s = t$. On the other hand, for the orienteering problem in the Euclidean space, Chen and Har-Peled [9] proposed a Polynomial Time Approximation Scheme (PTAS), which delivers a $(1-\epsilon)$-approximate solution within time $O(n^{\frac{1}{\epsilon}})$, where ϵ is a constant with $0 < \epsilon \leq 1$ and n is the number of nodes.

There are other studies on the team orienteering problem (TOP), where the objective is to find service tours for multiple vehicles and each node will be visited by no more than one vehicle. Boussier *et al.* [5] proposed an exact algorithm for the problem. Their algorithm is only applicable when the problem size is small, since the problem is NP-hard. Archetti *et al.* [1] devised a tabu search algorithm and a neighborhood search algorithm for the problem with exponential time complexity. Bianchessi *et al.* [3] proposed a branch-and-cut algorithm, and the algorithm could find better solutions for a given set of TOP problem instances. Vidal *et al.* [33] introduced a large neighborhood method with pruning and re-optimization techniques. Gavalas *et al.* [17] assumed that there is a profit of visiting a node only when the node is visited within its given time window. They proposed meta-heuristic approaches. Yu *et al.* [40] considered the factor that the profits of visiting a node at different time points are different, and they devised a bee colony algorithm. Hanafi *et al.* [19] considered a scenario where a node can be visited by different vehicles in a predefined order, and a profit for the node is received if the vehicles serve the node by the order. They proposed a kernel search framework for the team orienteering problem in the scenario. Orlis *et al.* [29] studied a problem of finding routes for vehicles to replenish cash to ATMs so that the number of bank account holders within a given distance of a replenished ATM is maximized, and devised an exact solution method based on column generation and a meta-heuristic based on large neighborhood search. Notice that the these meta-heuristic algorithms do not provide any performance guarantees on the solutions they delivered, and cannot apply for the case where a node may be visited multiple times and the profit of visiting the node is a nondecreasing submodular function of the number of visits.

We also notice that the submodular set function maximization problem is related to the GTOP problem. Nemhauser and Wolsey [28] considered the problem of maximizing a nondecreasing submodular set function under the constraint of choosing no more than K elements in a given set. They devised a greedy algorithm for it, which delivers a $(1 - 1/e)$-approximate solution, and that result is tight, where e is the base of the natural logarithm. They also extended their result to the submodular function maximization problem under the constraint of the intersection of P matriods, and proved that their greedy algorithm can find a $\frac{1}{P+1}$-approximate solution [16]. Filmus and Ward [15] improved the ratio to $1-1/e$ when $P = 1$. On the other hand, Buchbinder *et al.* [6] proposed a randomized approximation algorithm for the non-monotonically submodular function maximization problem without any constraints, and the expectation of the delivered solution is at least half the value of an optimal solution. However, they assumed that it takes polynomial time to find the element with the maximum marginal gain with respect to a partial solution. This assumption may not be realistic, since the orienteering problem is NP-hard.

III. Preliminaries

In this section, we introduce the system model and define the problem precisely.

A. System Model

Let $G = (V \cup \{s, t\}, E)$ be a given complete undirected graph, where $V = \{v_1, v_2, \ldots, v_n\}$ is a set of n to-be-served nodes, s is a source node, and t is a destination node. Notice that nodes s and t may or may not be co-located. There is an edge in E between any two nodes in $V \cup \{s, t\}$.

There are $K > 1$ vehicles to serve the nodes in V, and all vehicles are located at source node s initially. Each vehicle k needs to find a simple path P_k from node s to node t with $1 \leq k \leq K$.

The cost of path P_k for vehicle k is defined as follows, where the meaning of the 'cost' is the amount of energy consumed, or the amount of time elapsed, of the vehicle, depending on the application scenario. Let $P_k =< s,\ v_1,\ v_2, \cdots, v_{q_k},\ t >$, where q_k is the number of nodes served by vehicle k in P_k except nodes s and t, and $1 \leq k \leq K$. Fig. 3 illustrates the employment of $K = 2$ vehicles to serve nodes in a network.

We assume that the K vehicles are homogeneous. Denote by $c(v_i, v_{i+1})$ the cost of a vehicle k for traveling between nodes v_i and v_{i+1}, and $h(v_i)$ the service cost of vehicle k at node v_i. Assume that $h(s) = h(t) = 0$. The cost $w(P_k)$ of path P_k for vehicle k then is

$$w(P_k) = \sum_{i=1}^{q_k} h(v_i) + \sum_{i=0}^{q_k} c(v_i, v_{i+1}), \quad (1)$$

where $v_0 = s$ and $v_{q_k+1} = t$.

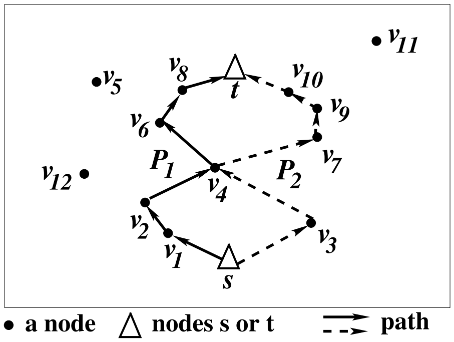


Fig. 3. Network model of the GTOP problem, where $K = 2$, path $P_1 =< s, v_1, v_2, v_4, v_6, v_8, t >$, $P_2 =< s, v_3, v_4, v_7, v_9, v_{10}, t >$, and nodes v_4 is contained in both P_1 and P_2.

Denote by B the cost budget of each vehicle k. The cost $w(P_k)$ of path P_k for vehicle k thus must be no greater than its cost budget B, i.e., $w(P_k) \leq B$.

B. Profit Function

For each node $v_i \in V$, denote by n_i the number of times that it is served by n_i vehicles among the K vehicles with $0 \leq n_i \leq K$. That is, v_i may be served more than once by different vehicles. We define the total profit received by serving the nodes in the K paths as follows.

We make use of a nondecreasing submodular function $u_i(n_i)$ to model the profit of serving node v_i by the n_i vehicles among the K vehicles. Function $u_i(.)$ has three properties. (i) $u_i(0) = 0$; (ii) the nondecreasing property: $0 \leq u_i(x) \leq u_i(y)$ if $0 \leq x \leq y$, where x and y are two integers; and (iii) the submodularity property: for any nonnegative integer Δ, $u_i(x + \Delta) - u_i(x) \geq u_i(y + \Delta) - u_i(y)$ if $0 \leq x \leq y$. This function characterizes the diminishing return received by serving node v_i with multiple times.

We here illustrate the physical meaning of function $u_i(n_i)$ with the following two examples. One example is that $u_i(n_i) = 1$ if $n_i \geq 1$; otherwise (i.e., $n_i = 0$), $u_i(n_i) = 0$. In this example, there is a profit of 1 if node v_i is served by at least one of the K vehicles; otherwise, the profit is 0. The other example is that $u_i(n_i) = \log_2(n_i + 1)$ [20], which implies that the more vehicles serve a node v_i, the less the marginal gain is obtained from the serving. In other words, this function is used to encourage visiting new nodes.

Notice that although both nodes s and t are contained in each of the K paths, we assume the profits for serving them are zeros, i.e., $u_s(n_s) = u_t(n_t) = 0$ with $0 \leq n_s, n_t \leq K$.

The total profit received from serving the nodes in the K paths $P_1, P_2, \ldots, P_K$ then is

$$\sum_{v_i \in \bigcup_{k=1}^{K} P_k} u_i(n_i), \tag{2}$$

where n_i is the number of times that v_i is served by the K vehicles.

C. Problem Definition

Given a graph $G = (V \cup \{s, t\}, E)$, *the generalized team orienteering problem (GTOP)* in G is to find K paths $P_1, P_2, \ldots, P_K$ for K vehicles with each starting from node s and ending at t, such that the profit sum of the nodes served by the K vehicles, i.e., $\sum_{v_i \in \bigcup_{k=1}^{K} P_k} u_i(n_i)$, is maximized, subject to that the cost $w(P_k)$ of each path P_k for vehicle k is no greater than the cost budget B, i.e., $w(P_k) \leq B$ with $1 \leq k \leq K$. That is,

$$\max \sum_{v_i \in \bigcup_{k=1}^{K} P_k} u_i(n_i), \tag{3}$$

subject to

$$w(P_k) \leq B, \quad 1 \leq k \leq K. \tag{4}$$

Formally, we use a binary decision variable x_{ik} to indicate whether node $v_i \in V$ is contained in path P_k, where $x_{ik} = 1$ if v_i is in P_k; Otherwise, $x_{ik} = 0$, for all i and k with $0 \leq i \leq n + 1$ and $1 \leq k \leq K$, where $v_0 = s$ and $v_{n+1} = t$. The number of paths in which v_i is contained is $n_i = \sum_{k=1}^{K} x_{ik}$. Similarly, we use an indicator decision variable y_{ijk} to indicate whether an edge (v_i, v_j) in E from v_i to v_j is contained in path P_k, where $y_{ijk} = 1$ if it is in path P_k; Otherwise, $y_{ijk} = 0$, where $0 \leq i, j \leq n + 1$ and $1 \leq k \leq K$.

The GTOP problem can then be formulated as follows.

$$\max_{x_{ik}, y_{ijk}} \sum_{v_i \in \bigcup_{k=1}^{K} P_k} u_i\left(\sum_{k=1}^{K} x_{ik}\right), \tag{5}$$

subject to

$$\sum_{i=1}^{n} x_{ik} \cdot h(v_i) + \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} y_{ijk} \cdot c(v_i, v_{i+1}) \leq B, \quad 1 \leq k \leq K \tag{6}$$

$$\sum_{j=1}^{n+1} y_{0jk} = 1, \quad \sum_{i=0}^{n} y_{i,n+1,k} = 1, \qquad 1 \leq k \leq K \tag{7}$$

$$\sum_{j=0, j \neq i}^{n+1} y_{jik} = \sum_{j=0, j \neq i}^{n+1} y_{ijk} = x_{ik}, \quad 1 \leq i \leq n, \quad 1 \leq k \leq K \tag{8}$$

$$\sum_{v_i, v_j \in S} y_{ijk} \leq |S| - 1, \quad 1 \leq k \leq K, \ \forall S \subset V, \ S \neq \emptyset \tag{9}$$

$$x_{ik} \in \{0, 1\}, \qquad 1 \leq i \leq n, \ 1 \leq k \leq K \tag{10}$$

$$x_{0k} = x_{n+1,k} = 1, \qquad 1 \leq k \leq K \tag{11}$$

$$y_{ijk} \in \{0, 1\}, \qquad 0 \leq i, j \leq n + 1, \ 1 \leq k \leq K, \tag{12}$$

where Constraint (6) ensures that the cost of each path P_k is no greater than the budget B of vehicle k; Constraint (7) ensures that nodes $v_0 (= s)$ and $v_{n+1} (= t)$ must be contained in each of the K paths; Constraint (8) indicates that each node v_i except nodes s and t has one incoming edge and one outgoing edge if it is contained in path P_k, and Constraint (9) implies that the number of edges with their endpoints contained in any nonempty proper subset S of V is no greater than $|S| - 1$, thus prevents disconnected closed subtours contained in a solution.

D. Approximation Ratio

Denote by OPT the value of an optimal solution of a maximization optimization problem. Also, denote by SOL

the value of a feasible solution delivered by an algorithm to the problem. The approximation ratio of the algorithm is α if $SOL \geq \alpha \cdot OPT$ for any problem instance, where $0 < \alpha \leq 1$.

IV. Approximation Algorithm for the Generalized Team Orienteering Problem

In this section, we propose a novel constant approximation algorithm for the GTOP problem.

The proposed algorithm proceeds iteratively. Within each iteration, one path with the maximum marginal gain is found for one vehicle in an auxiliary graph which will be constructed later. Thus, the algorithm has K iterations. The detailed algorithmic description is given as follows.

A. Algorithm

Given an undirected graph $G = (V \cup \{s,t\}, E)$, the algorithm first constructs an auxiliary graph $G' = (V \cup \{s,t\}, E; w' : E \mapsto \mathbb{R}^{\geq 0})$ from G, where the weight $w'(v_i, v_j)$ of each edge (v_i, v_j) in G' is

$$w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i) + h(v_j)}{2}, \tag{13}$$

$c(v_i, v_j)$ is the traveling cost between nodes v_i and v_j for a vehicle k, $h(v_i)$ and $h(v_j)$ are the service costs of vehicle k at nodes v_i and v_j, respectively. For any $s-t$ path P_k in G', denote by $w'(P_k)$ the weighted sum of the edges in P_k, i.e.,

$$w'(P_k) = \sum_{(v_i,v_j)\in P_k} w'(v_i, v_j). \tag{14}$$

There are two interesting relationships between graphs G and G', which are the corner stones of the proposed algorithm. That is,

(i) For any $s-t$ path P_k of vehicle k in G, the cost $w(P_k)$ of P_k in G (defined by Eq. (1)) is equal to the weighted sum $w'(P_k)$ of the edges in P_k of G' (defined by Eq. (14)).

(ii) The edge weights in G' satisfy the triangle inequality.

The proposed algorithm then finds K paths for the K vehicles. Assume that it has found k paths $P_1, P_2, \ldots, P_k$, where P_j is the path for vehicle j, $0 \leq k \leq K-1$, and $1 \leq j \leq k$. Also, assume that each node v_i in V has been served n_i times in the k service paths with $0 \leq n_i \leq k$. Initially, $k = 0$ and $n_i = 0$ for each $v_i \in V$. The algorithm now finds the $(k+1)$th path P_{k+1} for vehicle $k+1$ as follows.

The algorithm finds an approximate $s-t$ path P_{k+1} in G' under the cost budget B constraint, by applying an approximation algorithm for the orienteering problem, where the profit of serving a node v_i in G' is set to

$$u(v_i, k+1) = u_i(n_i+1) - u_i(n_i), \tag{15}$$

and v_i has already been served n_i times in the previous k paths. Denote by $u(P_{k+1})$ the profit sum of the nodes in path P_{k+1}, i.e., $u(P_{k+1}) = \sum_{v_i \in P_{k+1}} u(v_i, k+1)$.

After finding the $(k+1)$the path P_{k+1}, the algorithm updates the number of times n_i that v_i is served. The algorithm continues until the K paths are found.

The algorithm for the GTOP problem is presented in `Algorithm` 1.

Algorithm 1 Approximation Algorithm for the GTOP Problem (`approAlg`)

Require: $G = (V \cup \{s,t\}, E)$, K vehicles with the cost budget B, travel cost $c : E \mapsto \mathbb{R}^{\geq 0}$, service cost $h : V \mapsto \mathbb{R}^{\geq 0}$, and profit $u_i : \mathbb{Z}^{\geq 0} \mapsto \mathbb{R}^{\geq 0}$ for each $v_i \in V$.

Ensure: K $s-t$ paths such that the total profit for serving the nodes in the paths is maximized, subject to the cost budget constraints on the K paths.

Construct an auxiliary graph G' from G, where $G' = (V \cup \{s,t\}, E)$, $w' : E \mapsto \mathbb{R}^{\geq 0}$, and $w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i)+h(v_j)}{2}$ for each edge (v_i, v_j) in G';
Let $\mathcal{P} \leftarrow \emptyset$; /* the set of found paths */
Let $k \leftarrow 0$; /* the number of found paths */
Let $n_i \leftarrow 0$, for each $v_i \in V$; /* the number of times n_i that each node v_i has been served in the k paths */
while $k < K$ **do**
  Find an approximate $s-t$ path P_{k+1} in G' with cost budget B, by applying an approximation algorithm for the orienteering problem, where the profit for serving each node v_i in G' is set as $u(v_i, k+1) = u_i(n_i+1) - u_i(n_i)$;
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_{k+1}\}$;
  For each node v_i in path P_{k+1}, increase its number of served times n_i by one;
  $k \leftarrow k+1$;
end while
return the K paths in $\mathcal{P}$.

B. Algorithm Analysis

In the following, we first show that the optimization objective function is a nondecreasing submodular function. We then show that `Algorithm` 1 can find an approximate path P_k in graph G' for vehicle k. We finally analyze the approximation ratio of `Algorithm` 1.

We start with the following lemma.

Lemma 1: Given K $s-t$ paths $P_1, P_2, \ldots, P_K$ in the original graph G, let $\mathcal{P} = \{P_1, P_2, \ldots, P_K\}$, and $u(\mathcal{P}) = \sum_{v_i \in \bigcup_{k=1}^{K} P_k} u_i(n_i)$. Then, $u(\mathcal{P})$ is a nondecreasing submodular function.

Proof: The lemma can be easily shown, omitted. □

Lemma 2: `Algorithm` 1 can find an approximate $s-t$ path P_k in G' for vehicle k. Also, path P_k is a feasible path in the original graph G.

Proof: We first show that G' is a metric graph, i.e., its edge weights satisfy the triangle inequality. Consider any three nodes v_i, v_j, v_l in G', the edges formed by them are (v_i, v_j), (v_i, v_l), and (v_j, v_l), respectively. We have

$$\begin{aligned} & w'(v_i, v_j) \\ & = c(v_i, v_j) + \frac{h(v_i) + h(v_j)}{2}, \text{ by Eq. (13)} \\ & \leq c(v_j, v_l) + c(v_i, v_l) + \frac{h(v_i) + h(v_j)}{2}, \\ & \qquad \text{as the travel costs in } G \text{ satisfy the triangle inequality} \\ & \leq c(v_j, v_l) + c(v_i, v_l) + \frac{h(v_i) + h(v_j)}{2} + h(v_l) \end{aligned}$$

$$= c(v_j, v_l) + \frac{h(v_j)+h(v_l)}{2} + c(v_i, v_l) + \frac{h(v_i)+h(v_l)}{2}$$
$$= w'(v_j, v_l) + w'(v_i, v_l), \quad \text{by Eq. (13).} \tag{16}$$

That is, the edge weights in G' satisfy the triangle inequality.

We then can apply the approximation algorithm due to Chekuri *et al.* [8] for the orienteering problem in the metric graph G'.

The rest is to show that the cost $w(P_k)$ of any $s-t$ path P_k of vehicle k in G is equal to the weighted sum $w'(P_k)$ of the edges in P_k of G'.

Let $P_k = < s,\ v_1,\ v_2, \cdots, v_{q_k},\ t >$. Then,

$$\begin{aligned}
&w(P_k)\\
&= \sum_{i=1}^{q_k} h(v_i) + \sum_{i=0}^{q_k} c(v_i, v_{i+1}),\ \text{by Eq. (1)}\\
&= \frac{h(v_1)}{2} + \sum_{i=1}^{q_k-1}\frac{h(v_i)+h(v_{i+1})}{2} + \frac{h(v_{q_k})}{2} + \sum_{i=0}^{q_k} c(v_i, v_{i+1})\\
&= \frac{h(s)+h(v_1)}{2} + \sum_{i=1}^{q_k-1}\frac{h(v_i)+h(v_{i+1})}{2} + \frac{h(v_{q_k})+h(t)}{2}\\
&\quad + \sum_{i=0}^{q_k} c(v_i, v_{i+1}), \quad \text{as } h(s)=h(t)=0\\
&= \sum_{i=0}^{q_k}\left(\frac{h(v_i)+h(v_{i+1})}{2} + c(v_i, v_{i+1})\right),\ \text{as } v_0 = s,\ v_{q_k+1} = t\\
&= \sum_{i=0}^{q_k} w'(v_i, v_{i+1}) \ = \ w'(P_k), \quad \text{by Eq. (13).}
\end{aligned} \tag{17}$$

The lemma then follows. □

We finally analyze the approximation ratio of `Algorithm` 1 by the following theorem.

Theorem 1: Given a graph $G = (V \cup \{s,t\}, E)$, K vehicles with each vehicle having a cost budget B, the travel cost function $c : E \mapsto \mathbb{R}^{\geq 0}$, service cost function $h : V \mapsto \mathbb{R}^{\geq 0}$, and the profit function $u_i : \mathbb{Z}^{\geq 0} \mapsto \mathbb{R}^{\geq 0}$ for each node v_i in V, there is an approximation algorithm, `Algorithm` 1, for the GTOP problem, which delivers a $(1-(1/e)^\alpha)$-approximate solution, assuming that there is an α-approximation algorithm for the orienteering problem with $0 < \alpha < 1$, where e is the base of the natural logarithm.

Proof: Let paths $P_1^*, P_2^*, \ldots, P_K^*$ be the K paths in an optimal solution of the GTOP problem, and let $\mathcal{A}^*$ be the set of the K optimal paths, i.e., $\mathcal{A}^* = \{P_1^*, P_2^*, \ldots, P_K^*\}$. Also, let $\mathcal{A}_k^* = \{P_1^*, P_2^*, \ldots, P_k^*\}$ with $1 \leq k \leq K$.

Let $\mathcal{A}$ be the set of K paths delivered by `Algorithm` 1, i.e., $\mathcal{A} = \{P_1, P_2, \ldots, P_K\}$. Also, let $\mathcal{A}_k$ be the set of the first k paths among the K paths, i.e., $\mathcal{A}_k = \{P_1, P_2, \ldots, P_k\}$ with $1 \leq k \leq K$.

For any set $\mathcal{A}_k$, denote by $\hat{P}_{k+1}$ the optimal $s-t$ path in G', such that the marginal profit is maximized, i.e.,

$$\hat{P}_{k+1} = \arg\max_{P \in \mathcal{P}}\{u(\mathcal{A}_k \cup \{P\}) - u(\mathcal{A}_k)\}, \tag{18}$$

where $\mathcal{P}$ is the set of all feasible paths for the $(k+1)$th vehicle.

Notice that since the K vehicles are homogeneous, each P_j^* of the optimal K paths $P_1^*, P_2^*, \ldots, P_K^*$ is a feasible path for the $(k+1)$th vehicle, i.e., $\{P_1^*, P_2^*, \ldots, P_K^*\} \subset \mathcal{P}$.

Following Lemma 2, P_{k+1} is an α-approximate path found by `Algorithm` 1 with respect to $\hat{P}_{k+1}$, where $\hat{P}_{k+1}$ has the maximum marginal profit. We thus have

$$\begin{aligned}
u(\mathcal{A}_{k+1}) &- u(\mathcal{A}_k)\\
&= u(P_{k+1})\\
&\geq \alpha \cdot u(\hat{P}_{k+1})\\
&= \alpha \cdot (u(\mathcal{A}_k \cup \{\hat{P}_{k+1}\}) - u(\mathcal{A}_k)),
\end{aligned} \tag{19}$$

where $0 \leq k \leq K-1$.

We now consider the relationship between $u(\mathcal{A}_{k+1})$ and $u(\mathcal{A}_k)$ as follows. For each k and j with $0 \leq k \leq K-1$ and $1 \leq j \leq K$, we show that

$$u(\mathcal{A}_k \cup \mathcal{A}_j^*) \ \leq \ \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*) \tag{20}$$

as follows.

$$\begin{aligned}
&u(\mathcal{A}_k \cup \mathcal{A}_j^*)\\
&= u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^* \cup \{P_j^*\}),\ \text{by the definition of } \mathcal{A}_j^*\\
&= u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^* \cup \{P_j^*\}) - u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*)\\
&\leq u(\mathcal{A}_k \cup \{P_j^*\}) - u(\mathcal{A}_k) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*),\\
&\qquad \text{due to the submodularity of } u(.) \text{ and } \mathcal{A}_k \subseteq \mathcal{A}_k \cup \mathcal{A}_{j-1}^*\\
&\leq u(\mathcal{A}_k \cup \{\hat{P}_{k+1}\}) - u(\mathcal{A}_k) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*), \quad \text{as } P_j^* \in \mathcal{P}\\
&\qquad \text{and } \hat{P}_{k+1} \text{ is an optimal path in } \mathcal{P} \text{ with respect to } \mathcal{A}_k\\
&\leq \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*), \text{due to Eq. (19).}
\end{aligned}$$

We then have

$$\begin{aligned}
u(\mathcal{A}^*) &= u(\mathcal{A}_K^*), \quad \text{as } \mathcal{A}^* = \mathcal{A}_K^*\\
&\leq u(\mathcal{A}_k \cup \mathcal{A}_K^*), \quad \text{as function } u(.) \text{ is nondecreasing}\\
&\leq \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{K-1}^*),\\
&\qquad \text{due to Ineq. (20) where } j = K,\\
&\leq \frac{2}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{K-2}^*),\\
&\qquad \text{due to Ineq. (20) where } j = K-1,\\
&\ \vdots\\
&\leq \frac{K}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{K-K}^*),\\
&= \frac{K}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k),\\
&\qquad \text{where } \mathcal{A}_{K-K}^* = \mathcal{A}_0^* = \emptyset\\
&= \beta K(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k),\ \text{let } \beta = \frac{1}{\alpha}.
\end{aligned} \tag{21}$$

Re-arranging Ineq. (21), we have

$$\begin{aligned}
u(\mathcal{A}_{k+1}) &\geq \frac{\beta K - 1}{\beta K} u(\mathcal{A}_k) + \frac{u(\mathcal{A}^*)}{\beta K}\\
&= a \cdot u(\mathcal{A}_k) + b,
\end{aligned} \tag{22}$$

where $a = \frac{\beta K - 1}{\beta K}$ and $b = \frac{u(\mathcal{A}^*)}{\beta K}$ with $0 \leq k \leq K-1$.

We finally bound the total profit of the solution $\mathcal{A}_K = \{P_1, P_2, \ldots, P_K\}$ delivered by `Algorithm` 1 as follows.

$$u(\mathcal{A}_K) \geq a \cdot u(\mathcal{A}_{K-1}) + b,\ \text{by Eq. (22)}$$

$$
\begin{aligned}
&\geq a(a \cdot u(\mathcal{A}_{K-2}) + b) + b \\
&= a^2 \cdot u(\mathcal{A}_{K-2}) + ab + b \\
&\vdots \\
&\geq a^K \cdot u(\mathcal{A}_{K-K}) + b \sum_{k=0}^{K-1} a^k \\
&= b \sum_{k=0}^{K-1} a^k, \text{ as } \mathcal{A}_{K-K} = \emptyset \text{ and } u(\emptyset) = 0 \\
&= b \frac{1 - a^K}{1 - a} \\
&= \frac{u(\mathcal{A}^*)}{\beta K} \frac{1 - (\frac{\beta K - 1}{\beta K})^K}{1 - \frac{\beta K - 1}{\beta K}}, \\
&= u(\mathcal{A}^*)(1 - (1 - \frac{1}{\beta K})^K) \\
&= u(\mathcal{A}^*)(1 - ((1 - \frac{1}{\beta K})^{\beta K})^{\frac{1}{\beta}}) \\
&\geq u(\mathcal{A}^*)(1 - (1/e)^{\frac{1}{\beta}}), \quad \text{as } (1 - \frac{1}{\beta K})^{\beta K} \leq \frac{1}{e} \\
&= (1 - (1/e)^{\alpha}) u(\mathcal{A}^*), \quad \text{as } \alpha = \frac{1}{\beta}. \qquad (23)
\end{aligned}
$$

The theorem then follows. □

It must be mentioned that the analysis of Theorem 1 holds only for homogenous vehicles, not for heterogenous vehicles. This indicates that our corresponding claim in our previous conference version [39] is incorrect.

Corollary 1: There is a $(1-(1/e)^{\frac{1}{2+\epsilon}})$-approximation algorithm for the GTOP problem with a time complexity of $O(Kn^{O(1/\epsilon^2)})$, where e is the base of the natural logarithm, and ϵ is a given constant with $0 < \epsilon \leq 1$.

Proof: Following Chekuri *et al.* [8], there is an approximation algorithm for the orienteering problem, which finds a $\frac{1}{2+\epsilon}$-approximate $s-t$ path P with a cost budget B, where $0 < \epsilon \leq 1$. Due to Theorem 1, the approximation ratio is $1-(1/e)^{0.4} \approx 0.33$ when $\epsilon = 0.5$.

The analysis of the time complexity is as follows. Following Chekuri *et al.* [8], the running time of the $\frac{1}{2+\epsilon}$-approximation algorithm for the orienteering problem is $O(n^{O(1/\epsilon^2)})$. It can be seen from `Algorithm` 1 that the algorithm in [8] will be invoked K times. The time complexity of `Algorithm` 1 thus is $O(Kn^{O(1/\epsilon^2)})$. □

We note that the starting node s and the ending node t are co-located in many applications. In this case, we actually find closed tours for the K vehicles and we can obtain a better approximation ratio for this special case of the problem as follows.

Corollary 2: There is a $(1-1/\sqrt{e})$-approximation algorithm for the GTOP problem with a time complexity of $O(Kn^3 \log n)$, when the starting node s and ending node t of each path P_k of vehicle k are co-located, where e is the base of the natural logarithm and n is the number of nodes in G, where $1 \leq k \leq K$.

Proof: Since there is a $\frac{1}{2}$-approximation algorithm for the orienteering problem which is to find an s-rooted closed tour, due to Paul *et al.* [30], the approximation ratio of `Algorithm` 1 then is $1-1/\sqrt{e}$, which is no less than 0.39.

On the other hand, the time complexity of the $\frac{1}{2}$-approximation algorithm for the orienteering problem due to Paul *et al.* [30] is $O(n^3 \log n)$. The time complexity of `Algorithm` 1 thus is $O(Kn^3 \log n)$, as the algorithm in [30] will be invoked K times. □

V. Approximation Algorithm for a Special Case

In this section, we consider a special case of the generalized team orienteering problem, where (i) the starting node s and the ending node t are co-located (i.e., $s = t$); and (ii) the profit $u_i(n_i)$ that each node v_i is served once and multiple times is equal, that is, $u_i(0) = 0$, $u_i(1) = u_i(2) = \cdots = u_i(K)$. Such an example of this special profit function is that, every sensor can be charged by only one mobile charger, rather than by multiple chargers.

Formally, given a graph $G = (V \cup \{s\}, E)$, K vehicles, a cost budget B of each vehicle, a travel cost function $c : E \mapsto \mathbb{R}^{\geq 0}$, a node service cost function $h : V \mapsto \mathbb{R}^{\geq 0}$, and a profit function $u : V \mapsto \mathbb{R}^{\geq 0}$, *the team orienteering problem (TOP)* is to find K closed tours $C_1, C_2, \ldots, C_K$ with each containing node s, such that the profit sum of serving the nodes in the K closed tours, $\sum_{v_i \in \bigcup_{k=1}^K C_k} u(v_i)$, is maximized, subject to that the cost $w(C_k)$ of each tour C_k is no greater than B, where $1 \leq k \leq K$.

Although the proposed greedy algorithm in the previous section is applicable for the team orienteering problem with an approximation ratio of 0.39 by Corollary 2, we here devise an approximation algorithm for the problem, and we will show that the algorithm is able to find a better solution than that by the proposed greedy algorithm when the cost budget B of each vehicle is large.

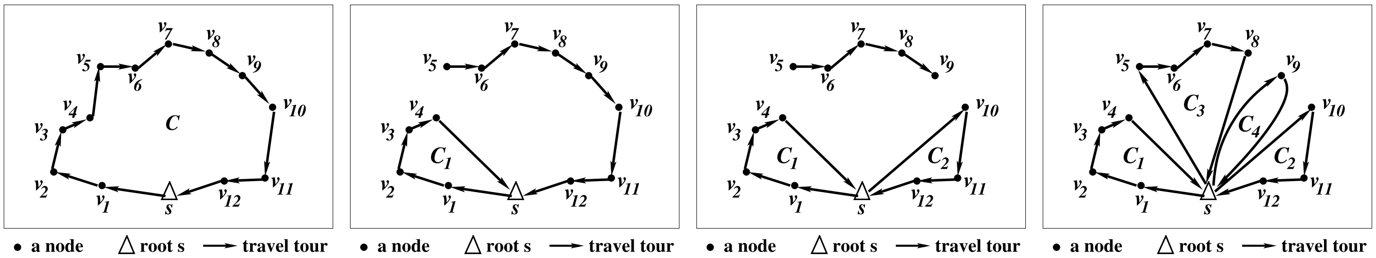
A. Algorithm

The basic idea behind the proposed algorithm is that it first finds an approximate closed tour C in G for the orienteering problem with a cost budget of KB, instead of B. It then splits the tour C into the minimum number of s-rooted tours $C_1, C_2, \ldots, C_{K'}$, subject to that the cost of each split tour is no greater than B, where K' is the number of tours split and $K' \geq K$. It finally chooses the top-K tours with the maximum profits among the K' tours.

Given a graph $G = (V \cup \{s\}, E)$, assume that the cost of each tour that serves only a single node v_i in V is no greater than B, i.e., $\max_{v_i \in V}\{h(v_i) + 2c(s, v_i)\} \leq B$. Otherwise (i.e., there is a node v_i in V such that $h(v_i) + 2c(s, v_i) > B$), node v_i can be removed from G, since it will not be contained in any feasible solution.

The algorithm proceeds as follows. It first constructs an auxiliary graph $G' = (V \cup \{s\}, E; w' : E \mapsto \mathbb{R}^{\geq 0})$ from G, where the weight $w'(v_i, v_j)$ of each edge (v_i, v_j) is $w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i)+h(v_j)}{2}$. It can be seen that the edge weights in G' satisfy the triangle inequality by Lemma 2.

The algorithm then finds a $\frac{1}{2}$-approximate s-rooted tour C for the orienteering problem in G' with a cost budget of KB, by applying an algorithm in work [30], where $u(v_i)$ is the profit of serving each node v_i in G'. Assume that $C = < s, v_1, v_2, \cdots, v_{n_C}, s >$, where n_C is the number of nodes



(a) tour C. (b) Obtained tour C_1 and the residual C (c) Obtained tours C_1 and C_2 and the residual C (d) Obtained $K' = 4$ tours C_1, C_2, C_3, and C_4 from C

Fig. 4. Illustration of the approximation algorithm for the team orienteering problem, where $K = 3$.

in C, e.g., see Fig. 4(a). Denote by $u(C)$ the profit sum of the nodes in C, i.e., $u(C) = \sum_{v_i \in C} u(v_i)$.

Having the tour C, the algorithm thirdly splits C into the minimum number of s-rooted tours $C_1, C_2, \ldots, C_{K'}$, subject to that the cost of each split tour is no greater than B, where K' is a positive integer determined as follows.

The first split tour is $C_1 =< s,\ v_1,\ v_2,\ \cdots,\ v_{l_1},\ s >$, where v_{l_1} is the last node along C such that the cost of C_1 is no greater than B (see Fig. 4(b)), which means the cost of tour $< s,\ v_1,\ v_2,\ \cdots,\ v_{l_1},\ v_{l_1+1},\ s >$ is strictly larger than B. The residual C is path $< v_{l_1+1},\ v_{l_1+2}, \cdots,\ v_{n_C},\ s >$ after splitting tour C_1 from C.

The second split tour is $C_2 =< s,\ v_{n_C},\ v_{n_C-1},\ \cdots,\ v_{l_2},\ s >$, where the v_{l_2} is the last node *backwards* along the residual C such that the cost of C_2 is no greater than B (see Fig. 4(c)), which means that the cost of tour $< s,\ v_{n_C},\ v_{n_C-1},\ \cdots,\ v_{l_2},\ v_{l_2-1},\ s >$ is strictly larger than B. The residual C is path $< v_{l_1+1},\ v_{l_1+2},\ \cdots,\ v_{n_{l_2-1}} >$ after splitting tour C_2.

The third split tour is $C_3 =< s,\ v_{l_1+1},\ v_{l_1+2},\ \cdots,\ v_{l_3},\ s >$, where v_{l_3} is the last node along the residual C such that the cost of C_3 is no greater than B (see Fig. 4(d)), which indicates that the cost of tour $< s,\ v_{l_1+1}, v_{l_1+2},\ \cdots,\ v_{l_3},\ v_{l_3+1},\ s >$ is strictly larger than B. The residual C is path $< v_{l_3+1},\ v_{l_3+2},\ \cdots,\ v_{n_{l_2-1}} >$ after splitting tour C_3. The split procedures of rest tours are similar to that of tour C_3. Let K' be the number of split tours in the end. Fig. 4(d) shows that $K' = 4$ tours C_1, C_2, C_3, C_4 have been split from tour C, where the last split tour C_4 consists of only nodes s and v_9.

Having split K' tours $C_1, C_2, \ldots, C_{K'}$, let $u(C_k)$ be the profit of tour C_k, which is the profit sum of the nodes in tour C_k, i.e.,

$$u(C_k) = \sum_{v_i \in C_k} u(v_i), \tag{24}$$

where $1 \le k \le K'$.

For the sake of convenience, we assume that $u(C_1) \ge u(C_2) \ge \cdots \ge u(C_{K'})$. The algorithm finally chooses the top-K tours with the largest profits among the K' tours, i.e., $C_1, C_2, \ldots, C_K$, as the solution to the team orienteering problem if $K \le K'$. For example, the algorithm chooses tours C_1, C_2, and C_3 as the solution, since the profit of tour C_4 is the smallest. Otherwise ($K > K'$), the K' tours form the solution to the problem.

The algorithm for the team orienteering problem is presented in `Algorithm` 2.

Algorithm 2 Approximation Algorithm for the Team Orienteering Problem (`approAlgSpecial`)

Require: $G = (V \cup \{s\}, E)$, K vehicles with cost budget B, travel cost $c : E \mapsto \mathbb{R}^{\ge 0}$, service cost $h : V \mapsto \mathbb{R}^{\ge 0}$, and profit $u : V \mapsto \mathbb{R}^{>0}$.

Ensure: K s-rooted closed tours such that the total profit for serving the nodes in the tours in maximized, subject to the cost budget constraints on the K tours.

Construct an auxiliary graph $G' = (V \cup \{s\}, E; w' : E \mapsto \mathbb{R}^{\ge 0})$ from G, where $w'(v_i, v_j) = c(v_i, v_j) + \frac{h(v_i)+h(v_j)}{2}$ for each edge (v_i, v_j) in G';

Find a $\frac{1}{2}$-approximate tour C for the orienteering problem in G' with cost budget KB, by invoking an algorithm from [30];

Split tour C into, say K', s-rooted tours $C_1, C_2, \ldots, C_{K'}$, such that the cost of each split tour is no greater than B;

Let $\mathcal{C}$ be the set of the K tours with the maximum profits among the K' tours;

return the K tours in $\mathcal{C}$.

B. Algorithm Analysis

We assume that $K \ge 2$. Otherwise (i.e., $K = 1$), the team orienteering problem degenerates to the orienteering problem. In the following, we first obtain a nontrivial upper bound on the team orienteering problem. We then analyze the approximation ratio of the proposed algorithm.

Assume that the optimal solution contains K tours $C_1^*, C_2^*, \ldots, C_K^*$. Denote by OPT the optimal value, i.e., $OPT = \sum_{v_i \in \bigcup_{k=1}^{K} C_k^*} u(v_i)$. Meanwhile, denote by C_L^* the optimal solution to the orienteering problem in G' with the cost budget KB.

We start with the following important lemma.

Lemma 3: The optimal value OPT of the team orienteering problem in G is no greater than the value of the optimal solution C_L^* to the orienteering problem in G' with cost budget KB, i.e., $OPT \le u(C_L^*)$.

Proof: Consider the optimal solution consisting of the K tours $C_1^*, C_2^*, \ldots, C_K^*$. Since each tour contains the root s, a tour C that visits root s and the nodes in the K tours can be constructed, such that the profit $u(C)$ of tour C is the profit sum of the nodes in the K tours, and the cost $w(C)$ of tour C is no greater than KB, as the cost of each of the K tours is no larger than B. It then can be seen that tour C is a feasible solution to the orienteering problem in G' with the cost budget KB. Since C_L^* is the optimal solution, we have

$$OPT = \sum_{v_i \in \bigcup_{k=1}^{K} V(C_k^*)} u(v_i) \;=\; u(C) \;\le\; u(C_L^*). \quad (25)$$

The lemma then follows. □

We then analyze the approximation ratio of Algorithm 2, by distinguishing into two cases: (i) the number K' of split tours in Algorithm 2 is no more than K, i.e., $K' \le K$; and (ii) $K' > K$.

Theorem 2: Algorithm 2 delivers a $\frac{1}{2}$-approximate solution to the team orienteering problem in G if $K' \le K$. Otherwise ($K' > K \ge 2$), it delivers an α-approximate solution, where $\alpha = \frac{K}{2\lceil \frac{KB}{B-2\Delta} \rceil}$, B is the cost budget of each vehicle, $\Delta = \frac{1}{2}\max_{v_i \in V}\{2c(s, v_i) + h(v_i)\}$ is half the maximum cost for serving a node in V, and $\Delta \le \frac{B}{2}$. In addition, the time complexity of Algorithm 2 is $O(n^3 \log n)$, where n is the number of nodes in G.

Proof: We first consider case (i) that $K' \le K$. Since $K' \le K$, the solution delivered by Algorithm 2 consists of the K' split tours $C_1, C_2, \ldots, C_{K'}$. We have that

$$\begin{aligned}\sum_{v_i \in \bigcup_{k=1}^{K'} C_k} u(v_i) \;&=\; u(C)\\ &\ge \frac{1}{2}\cdot u(C_L^*), \text{ as } C \text{ is a } \frac{1}{2}\text{-approximate solution}\\ &\ge \frac{1}{2}\cdot OPT, \text{ by Lemma 3.} \end{aligned} \quad (26)$$

We then consider case (ii) that $K' > K \ge 2$ as follows. We first bound the number K' of split tours by Algorithm 2. It can be seen that $\Delta = \max_{v_i \in V}\{w'(s, v_i)\}$.

Recall that the cost $w'(C)$ of tour C is no greater than KB. After splitting off C_1 from C, the cost of the residual C is no more than $KB - (B - \Delta) = (K-1)B + \Delta$, as the cost of tour $< s,\ v_1,\ v_2,\ \cdots,\ v_{l_1},\ v_{l_1+1},\ s >$ is strictly larger than B and the cost of edge (v_{l_1+1}, s) is no more than Δ. Similarly, after splitting off C_2, the cost of the residual C is no greater than $(K-1)B + \Delta - (B - \Delta) = (K-2)B + 2\Delta$. Furthermore, after splitting off C_3, the cost of the residual C is no more than $(K-2)B + 2\Delta - (B - 2\Delta)$, since the cost of tour $< s,\ v_{l_1+1},\ v_{l_1+2},\ \cdots,\ v_{l_3},\ v_{l_3+1},\ s >$ is strictly larger than B, and the costs $w(s, v_{l_1+1})$ and $w(v_{l_3+1}, s)$ of both edges (s, v_{l_1+1}) and (v_{l_3+1}, s) are no more than Δ. That is, the cost the residual C will be reduced by at least $(B - 2\Delta)$ for splitting off each of the tours $C_3, C_4, \ldots, C_{K'-1}$ except the last tour $C_{K'}$. Thus, the number K' of split tours from C is upper bounded by

$$\begin{aligned}K' &\le 2 + \lceil \frac{(K-2)B + 2\Delta}{B - 2\Delta} \rceil\\ &= \lceil \frac{KB - 2\Delta}{B - 2\Delta} \rceil \;\le\; \lceil \frac{KB}{B - 2\Delta} \rceil. \end{aligned} \quad (27)$$

Since Algorithm 2 chooses the top-K tours with the largest profits among the K' split tours and $K \le K'$, the profit of the chosen K tours should be no less than $\frac{K}{K'}$ of the profit of the K' tours, i.e., $u(\mathcal{C}) \ge \frac{K}{K'} \cdot \sum_{v_i \in \bigcup_{k=1}^{K'} C_k} u(v_i)$.

The ratio of $u(\mathcal{C})$ to OPT thus is

$$\begin{aligned}\frac{u(\mathcal{C})}{OPT} &\ge \frac{\frac{K}{K'}\cdot \sum_{v_i \in \bigcup_{k=1}^{K'} C_k} u(v_i)}{OPT}\\ &\ge \frac{\frac{K}{K'}\cdot \frac{1}{2}OPT}{OPT}, \text{ by Ineq. (26)}\\ &= \frac{1}{2}\cdot\frac{K}{K'}\\ &\ge \frac{1}{2}\cdot\frac{K}{\lceil \frac{KB}{B-2\Delta} \rceil}, \text{ by Ineq. (27).}\end{aligned} \quad (28)$$

Remark: We compare the approximation ratio $\frac{K}{2\lceil \frac{KB}{B-2\Delta} \rceil}$ of Algorithm 2 against the ratio $(1 - 1/\sqrt{e})$ of Algorithm 1. On one hand, the value of $(1 - 1/\sqrt{e})$ is no more than 0.4. On the other hand, assume that the value of $\lceil \frac{KB}{B-2\Delta} \rceil$ can be approximated by $\frac{KB}{B-2\Delta}$, when the value of K is sufficiently large. Then, the approximation ratio $\frac{K}{2\lceil \frac{KB}{B-2\Delta} \rceil} \approx \frac{K}{2\frac{KB}{B-2\Delta}} = \frac{B-2\Delta}{2B} \ge 0.4 \ge 1 - 1/\sqrt{e}$, if $B \ge 10\Delta$. Notice that in many applications, the cost budget B of each vehicle usually is much larger than the maximum cost 2Δ of serving only a single node, as a vehicle can serve many nodes with its cost budget.

We finally analyze the time complexity of Algorithm 2. The construction of graph G' takes time $O(m)$, where $m = |E|$. The invoking of the algorithm in [30] takes time $O(n^3 \log n)$, where $n = |V|$. Notice that the tour splitting will take time only $O(n)$. Therefore, the time complexity of Algorithm 2 is $O(m) + O(n^3 \log n) + O(n) = O(n^3 \log n)$. □

VI. Performance Evaluation

In this section we evaluate the performance of the proposed algorithms. We also study the impact of important parameters, including the number of vehicles, the network size, and the cost budgets of vehicles, on the performance of the proposed algorithms.

A. Experimental Settings

We consider three applications of the GTOP problem: the first application is to schedule UAVs to monitor PoIs in a disaster area; the second one is to dispatch multiple mobile chargers to charge sensors in a sensor network; and the final one consists of classic benchmark instances of the team orienteering problem but the starting node s and ending node t are different, where the optimal solutions (or near-optimal solutions) to the instances are known. The main difference of the three applications is that, the profit of monitoring a PoI by multiple UAVs is larger than that by a single UAV in the first application [20], while the profits of serving a node once and multiple times are the same in the second and third applications [33], [36]. In the following, we describe the experimental settings of the three applications, respectively.

1) Application A: Employing UAVs to Monitor PoIs: We consider a disaster area with from 50 to 200 PoIs in a 2 km×2 km square [20]. The number of UAVs K varies from 1 to 5, which stay at a depot s initially and the depot is located at the center of the square. The energy capacity B of each UAV is 89.2 Wh [12]. Also, the flying energy consumption of each UAV between any two PoIs v_i and v_j is $c(v_i, v_j) = \eta \cdot d_{ij}$, where η is the energy consumption on flying per unit length with $\eta = 17.84$ J/m [12], and d_{ij} is their Euclidean distance. In addition, the energy consumption $h(v_i)$ for monitoring a PoI v_i is randomly chosen from an interval [500 J, 1,000 J]. Finally, the profit of monitoring a PoI v_i is $u(v_i) = p_i \cdot \log_2(n_i + 1)$ [20], where n_i is the times visited by the K UAVs, p_i is the priority of v_i, and p_i is randomly chosen from an interval $[p_{min},\ p_{max}]$ with $p_{min} = 1$ and $p_{min} = 5$.

2) Application B: Dispatching Chargers to Charge Sensors: There are from 50 to 200 sensors deployed in a 1 km×1 km square area randomly [25]. The battery capacity of each sensor v_i is 10.8 kJ [22]. Also, the residual energy re_i of sensor v_i is randomly chosen from an interval [0, 10.8 kJ], and the profit $u_i(v_i)$ of recharging sensor v_i is the amount of energy charged to it, i.e., $u_i(v_i) = 10.8\ kJ - re_i$. It can be seen that the profit of recharging one sensor with less residual energy is larger than that of recharging the other sensor with much residual energy.

A depot s is located at the center of the square area, and K mobile chargers are located at the depot initially, where the value of K varies from 1 to 8. We assume that the energy capacity B of each mobile charger is 2,000 kJ [21]. Also, the traveling energy consumption of a charger between any two nodes v_i and v_j is $c(v_i, v_j) = \eta \cdot d_{ij}$, where η is the energy consumption on traveling per unit length with $\eta = 0.6$ kJ/m [21], and d_{ij} is the Euclidean distance between v_i and v_j. On the other hand, the amount of energy consumption on recharging sensor v_i by a charger is $h(v_i) = \frac{10.8\ kJ - re_i}{\rho}$, where $10.8\ kJ - re_i$ is the amount of energy charged to v_i, and ρ is the energy charging efficiency of a charger with $\rho = 0.95$.

3) Application C: Benchmark Instances: We consider the 157 network instances given by [7], [33], where the starting node s and ending node t are different. The 157 instances are classified into four sets 4, 5, 6, and 7 [33], where the network topologies of different instances in the same set are identical, but the numbers of vehicles K (from 2 to 4) and the capacities of each vehicle B of the different instances are different. The number of nodes in the four sets are 100, 66, 64, and 102, respectively. Each node except nodes s and t is allowed to be visited no more than once.

All datasets used in this article are available at the website: https://zenodo.org/record/4047986#.X21VvIu-u2u.

B. Benchmark Algorithms

We here consider two algorithms for the benchmark purpose against the proposed algorithms `approAlg` and `approAlgSpecial`.

(1) Algorithm `partitionAlg` [34] first sorts all nodes in anticlockwise order with centering at node s. Assume that $v_1, v_2, \ldots, v_n$ is the order of sorted nodes. The algorithm then partitions the nodes into K disjoint sets $V_1, V_2, \ldots, V_K$, by the energy capacities of the K vehicles. That is, nodes $v_1, v_2, \ldots, v_{n_1}$ are in set V_1, where $n_1 = \lceil n \cdot \frac{1}{K} \rceil$; nodes $v_{n_1+1}, v_{n_1+2}, \ldots, v_{n_2}$ are in set V_2, where $n_2 = \lceil n \cdot \frac{2}{K} \rceil$; $\ldots$; nodes $v_{n_{K-1}+1}, v_{n_{K-1}+2}, \ldots, v_{n_K}$ are in set V_K, where $n_K = \lceil n \cdot \frac{K}{K} \rceil = n$. After partitioning the nodes, the algorithm finds an $s - t$ path P_k of vehicle k for serving the nodes in each set V_k by applying the approximation algorithm for the orienteering problem [8].

(2) Algorithm `forestAlg` [1] finds the K paths by starting with a forest with K trivial trees and each tree T_k consists of an edge (s, t) between the starting nodes s and the ending node t initially. Assume that some nodes have been inserted to the forest already.

For each node v_i, assume that it has been contained in n_i of the K trees with $0 \leq n_i \leq K$. The algorithm calculates both the marginal profit $\pi_i = u_i(n_i + 1) - u_i(n_i)$ and the increased cost δ_i of serving v_i, where the increased cost of serving v_i is the smallest difference between the cost of the $s - t$ path P_k visiting nodes in T_k and the cost of the $s - t$ path P'_k visiting both nodes in T_k and v_i, subject to the constraints that the cost of P'_k is no larger than B and v_i was not contained in T_k, i.e., $\delta_i = \min_{k=1, v_i \notin T_k,\ w(P'_k) \leq B}^{K} \{w(P'_k) - w(P_k)\}$. Notice that given a tree T_k containing s and t, an $s - t$ path P_k visiting nodes in T_k can be derived as follows. First, obtain a graph G'' by replicating edges in T_k except the edges on the path from s to t in T_k. Then, find a Eulerian path from s to t in graph G''. Finally, shortcut repeated nodes in the Eulerian path. It can be seen that the cost of path P_k is no more than twice the cost of tree T_k. The algorithm then inserts a node v_j to the forest with the maximum ratio of the marginal profit π_j to the increased cost δ_j of serving node v_j, i.e., $v_j = \arg\max_{v_i \in V} \{\frac{\pi_i}{\delta_i}\}$. This procedure of the forest growth continues until the insertion of any node will violate the cost budgets of the K vehicles.

The value in each figure is the average of the results out of 50 problem instances with the same network size. The running time of each algorithm is obtained based on a server with a 2.7 GHz Intel i7 CPU and an 8 GB RAM.

C. Algorithm Performance in the Application of Employing UAVs to Monitor PoIs

We first investigate the algorithm performance, by increasing the UAV energy capacity B from 50 Wh to 100 Wh when there are $n = 100$ PoIs and $K = 2$ UAVs in the network. Fig. 5 demonstrates that the profit sum by algorithm `approAlg` is around from 4% to 48% larger than that by algorithm `partitionAlg`, and is from 8% to 14% larger than that by algorithm `forestAlg`. It also can be seen that the profit sum by algorithm `partitionAlg` will not increase when the UAV energy capacity B is over 70 Wh, as each PoI will be monitored no more than once by UAVs in the solution delivered by the algorithm.

We then study the performance of the proposed by increasing the number of UAVs K from 1 to 5 in a network with $n = 100$ PoIs. Fig. 6 shows that the profit sum by algorithm `approAlg` is about 10% higher than that by algorithm

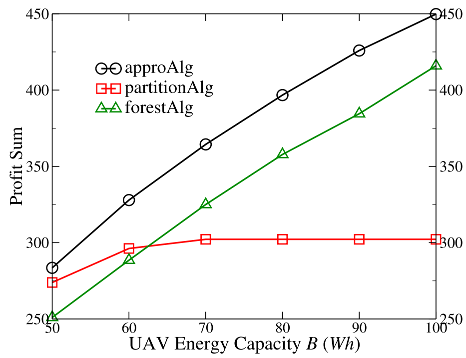


Fig. 5. Performance of different algorithms by varying the UAV energy capacity B from 50 Wh to 100 Wh, when $n = 100$ and $K = 2$.

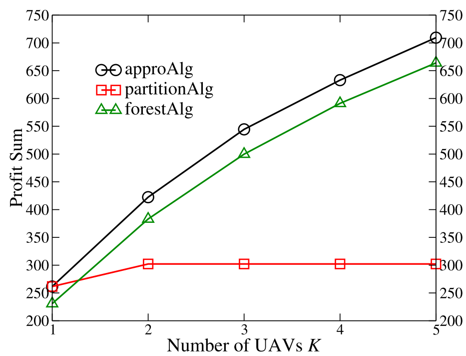


Fig. 6. Performance of different algorithms by varying the number of UAVs K from 1 to 5 when there are $n = 100$ PoIs.

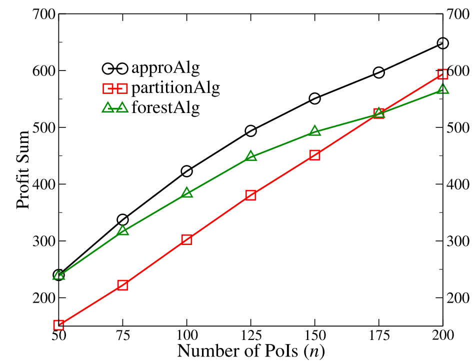


Fig. 7. Performance of different algorithms by varying the number of PoIs n from 50 to 200 when $K = 2$.

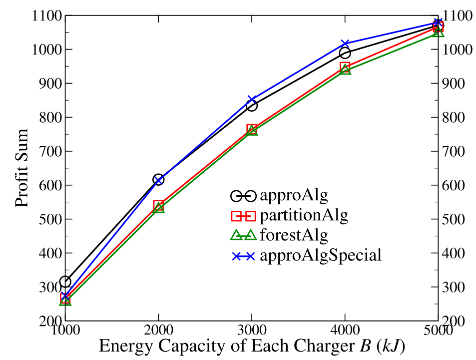


Fig. 8. Performance of different algorithms by varying the energy capacity B from 1,000 kJ to 5,000 kJ when $n = 200$ and $K = 2$.

forestAlg, and is significantly larger than that by algorithm partitionAlg when there are more than two UAVs in the network. For example, the profits sums by algorithms approAlg, partitionAlg and forestAlg are 422, 302, and 383, respectively, when $K = 2$. The rationale behind is that each PoI will be monitored no more than once in algorithm partitionAlg, while the PoI may be monitored multiple times by different UAVs in both algorithms forestAlg and approAlg, where the profit of monitoring the PoI has a diminishing return with multiple visits. Fig. 6 also plots an interesting phenomenon, that is, that the profit sums by algorithms approAlg and partitionAlg are equal when only one UAV is employed, as the GTOP problem degenerates to the traditional orienteering problem when $K = 1$ and they thus deliver the same solution.

We also evaluate the performance of different algorithms by increasing the number of PoIs n from 50 to 200 when there are $K = 2$ UAVs. Fig. 7 shows that the profit sum of the tours delivered by the proposed algorithm approAlg is about from 9% to 58% higher than that by algorithm partitionAlg, and is around from 1% to 15% higher than that by algorithm forestAlg. For example, the profit sums by algorithms approAlg, partitionAlg and forestAlg are 550, 451, and 492, respectively, when there are $n = 150$ PoIs. Fig. 7 demonstrates that the profit sum by each of the three mentioned algorithms increases with the growth of the number n of PoIs. The rationale behind is that PoIs are more densely located in a larger network, and the energy consumption of a UAV on its flying between different PoIs thus becomes smaller. Therefore, each UAV has more energy to monitor more PoIs.

D. Algorithm Performance in the Application of Dispatching Multiple Mobile Chargers to Charge Sensors

We first evaluate the performance of different algorithms, by varying the energy capacity B of each charger from 1,000 kJ to 5,000 kJ when there are $n = 200$ sensors and $K = 2$ chargers in the network. Fig. 8 shows that the profit sum by algorithm approAlgSpecial is slightly smaller than that by algorithm approAlg when the energy capacity B of each charger is no more than 2,000 kJ, while the profit sum by the former algorithm is up to 4% larger than that by the latter when B is larger than 2,000 kJ, which validates our claim that the approximation ratio of algorithm approAlgSpecial is larger than that of algorithm approAlg when the value of B is large (also see the analysis of Theorem 2 in Section V-B). Fig. 8 also demonstrates that the profit sums by algorithms approAlg and approAlgSpecial are from 3% to 16% higher than those by algorithms partitionAlg and forestAlg.

We then investigate the algorithm performance, by increasing the number of mobile chargers K from 1 to 8 in a network with $n = 200$ sensors. Fig. 9 demonstrates that the profit sums by algorithms approAlg and approAlgSpecial is up to 16% larger than those by algorithms partitionAlg and forestAlg. Fig. 9 illustrates two interesting phenomenons: one is that the profit sums by algorithms approAlg,

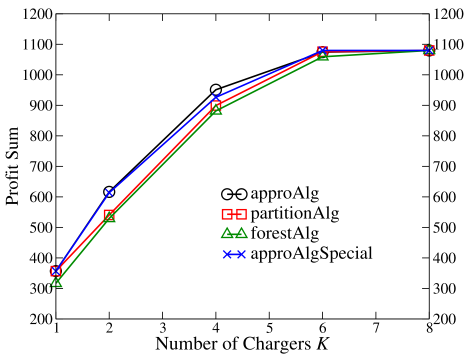


Fig. 9. Performance of different algorithms by varying the number of chargers K from 1 to 10 when $n = 200$ and $B_{max} = 1,500$ kJ.

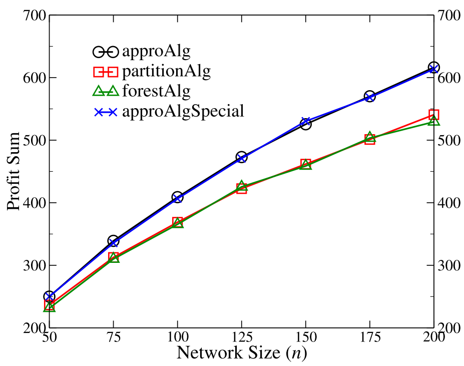
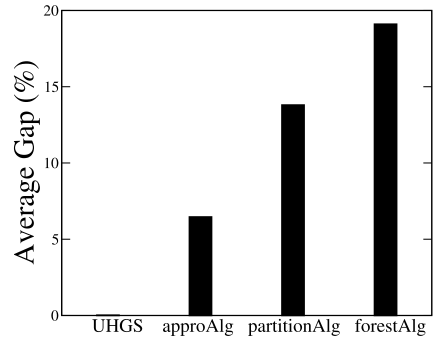
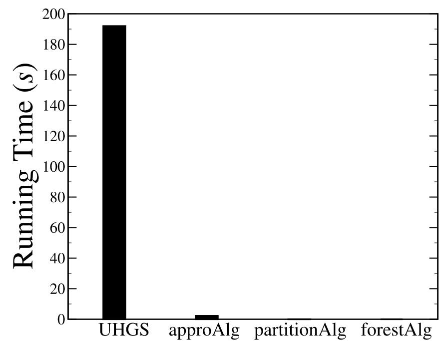


Fig. 10. Performance of different algorithms by varying the network size n from 50 to 200, when $K = 2$ and $B = 2,000$ kJ.



(a) Average gap



(b) Algorithm running time

Fig. 11. Performance of different algorithms in the 157 benchmark instances [33].

`approAlgSpecial` and `partitionAlg` are equal when one ($K = 1$) mobile charger is employed, as the GTOP problem degenerates to the traditional orienteering problem when $K = 1$; another is that the profit sums by the four algorithms are close to each other when $K = 8$ mobile chargers are employed, since almost all sensors are charged in the charging tours delivered by each of the four algorithms when $K = 8$ chargers are employed.

We further study the performance of different algorithms by varying the network size n from 50 to 200 when there are $K = 2$ mobile chargers in the sensor network. Fig. 10 shows that the profit sums of the tours delivered by the proposed algorithms `approAlg` and `approAlgSpecial` are about from 5% to 16% higher than those by algorithms `partitionAlg` and `forestAlg`, respectively. For example, the profit sums by algorithms `approAlg`, `approAlgSpecial`, `partitionAlg` and `forestAlg` are 616, 613, 540, and 529, respectively, when the network size n is 200.

E. Algorithm Performance With the Benchmark Instances

We finally study the performance of different algorithms with the 157 benchmark instances [33]. In addition to algorithms `approAlg`, `partitionAlg`, and `forestAlg`, we also consider a heuristic algorithm `UHGS` [33] that can find a near-optimal solution. Notice that the quality of a solution delivered by an algorithm is measured by a percentage of the deviation from the best-known solution (BKS) so far with expensive computation [33], which is $100(1 - \frac{u}{u_{BSK}})$, where u is the profit of the solution and u_{BKS} is the profit of the BKS.

Experimental results showed that algorithm `approAlg` can find 8 optimal solutions for the 157 instances, while algorithm `UHGS` delivers 128 optimal solutions. Notice that the objective of this article is not to find optimal solutions to the team orienteering problem, as the problem is NP-hard. Instead, the goal of this article is to find performance-guaranteed solutions within a short time.

On the other hand, Fig. 11(a) shows that the average performance gaps of algorithms `UHGS` and `approAlg`, which are about 0.02% and 6.4%, respectively, outperform the average performance gaps of algorithms `partitionAlg` and `forestAlg`. Fig. 11(a) also demonstrates that the solution delivered by algorithm `approAlg` is about 93.6% (=1-6.4%) of the optimal solution, which is much better, compared with its analytical approximation ratio $1 - (1/e)^{\frac{1}{2+\epsilon}} = 1 - (1/e)^{\frac{1}{2+0.5}} \approx 0.33$ with $\epsilon = 0.5$. This clearly indicates that the estimate on the theoretical approximation ratio 0.33 is very conservative.

Fig. 11(b) compares the running times of different algorithms, from which it can be seen that the running time of algorithm `UHGS` is quite long, as high as 192 seconds, while the running time of algorithm `approAlg` only takes about 2.3 seconds.

Although the performance of algorithm `UHGS` is better than that of the proposed algorithm `approAlg`, it may be more practical to apply algorithm `approAlg` in real applications, as it can find a performance-guaranteed solution within an acceptable time frame. For example, in the application of employing UAVs to monitor PoIs in a disaster area, it is urgent to find high-quality flying tours for the UAVs in a short time. Otherwise, if a long time is needed to find high-quality flying tours, the waiting time of some PoIs before they are monitored by a UAV may be prohibitively long, people who are trapped at the PoIs may have already been dead when the PoIs are monitored later. In addition, algorithm `approAlg` is applicable to scenarios in which each node can be visited by multiple vehicles, and the profit of monitoring the node is a submodular function of the number of vehicles visiting it. However, algorithm `UHGS` serves as a benchmark as it can deliver a near-optimal solution at the expense of the excessive amount of running time.

VII. Conclusion

In this article we studied the generalized team orienteering problem, where each node can be served by multiple vehicles and the profit of serving the node is a submodular function of the number of vehicles serving it. We proposed a novel $(1-(1/e)^{\frac{1}{2+\epsilon}})$-approximation algorithm for the problem, where ϵ is a given constant with $0 < \epsilon \leq 1$ and e the base of the natural logarithm. Particularly, the approximation ratio is about 0.33 when $\epsilon = 0.5$. For a special case of the problem where the profit of serving a node is the same no matter whether it is served only once or multiple times, we proposed an improved approximation algorithm with a better approximation ratio. We finally evaluated the proposed algorithms with simulations, and experimental results showed that the proposed algorithms are very promising.

Acknowledgment

The authors appreciate the three anonymous referees and the associate editor for their constructive comments and valuable suggestions, which helped them to improve the quality and presentation of the paper greatly.

References

[1] C. Archetti, A. Hertz, and M. G. Speranza, "Metaheuristics for the team orienteering problem," *J. Heuristics*, vol. 13, no. 1, pp. 49–76, Jan. 2007.

[2] N. Bansal, A. Blum, S. Chawla, and A. Meyerson, "Approximation algorithms for deadline-TSP and vehicle routing with time-windows," in *Proc. 36th Annu. ACM Symp. Theory Comput. (STOC)*, 2004, pp. 166–174.

[3] N. Bianchessi, R. Mansini, and M. G. Speranza, "A branch-and-cut algorithm for the team orienteering problem," *Int. Trans. Oper. Res.*, vol. 25, no. 2, pp. 627–635, Mar. 2018.

[4] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, "Approximation algorithms for orienteering and discounted-reward TSP," *SIAM J. Comput.*, vol. 37, no. 2, pp. 653–670, Jan. 2007.

[5] S. Boussier, D. Feillet, and M. Gendreau, "An exact algorithm for team orienteering problems," *4OR*, vol. 5, no. 3, pp. 211–230, Sep. 2007.

[6] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz, "A tight linear time (1/2)-approximation for unconstrained submodular maximization," *SIAM J. Comput.*, vol. 44, no. 5, pp. 1384–1402, Jan. 2015.

[7] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *Eur. J. Oper. Res.*, vol. 88, no. 3, pp. 464–474, Feb. 1996.

[8] C. Chekuri, N. Korula, and M. Pál, "Improved algorithms for orienteering and related problems," *ACM Trans. Algorithms*, vol. 8, no. 3, p. 23, Jul. 2012.

[9] K. Chen and S. Har-Peled, "The orienteering problem in the plane revisited," in *Proc. 22nd Annu. Symp. Comput. Geometry (SCG)*, 2006, pp. 247–254.

[10] H. Dai, H. Ma, A. X. Liu, and G. Chen, "Radiation constrained scheduling of wireless charging tasks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 1, pp. 314–327, Feb. 2018.

[11] L. Deng *et al.*, "Approximation algorithms for the min-max cycle cover problem with neighborhoods," *IEEE/ACM Trans. Netw.*, vol. 28, no. 4, pp. 1845–1858, Aug. 2020.

[12] (2020). *DJI Phantom 4 Pro*. [Online]. Available: https://www.dji.com/cn

[13] (2017). *Drone Using Hurricane IRMA*. [Online]. Available: http://thedronegirl.com/2017/09/16/7-ways-drones-helping-hurricane-irmaharvey-recovery-experts/

[14] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervas. Comput.*, vol. 16, no. 1, pp. 24–32, Jan. 2017.

[15] Y. Filmus and J. Ward, "A tight combinatorial algorithm for submodular maximization subject to a matroid constraint," in *Proc. IEEE 53rd Annu. Symp. Found. Comput. Sci.*, Oct. 2012, pp. 659–668.

[16] M. L. Fisher, G. L. Nemhausert, and L. A. Wolsey, "An analysis of the approximations for maximizing submodular set functions—II," *Math. Program. Study*, vol. 8, pp. 73–87, 1978.

[17] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and N. Vathis, "Efficient metaheuristics for the mixed team orienteering problem with time windows," *Algorithms*, vol. 9, no. 1, p. 6, Jan. 2016.

[18] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *Eur. J. Oper. Res.*, vol. 255, no. 2, pp. 315–332, Dec. 2016.

[19] S. Hanafi, R. Mansini, and R. Zanotti, "The multi-visit team orienteering problem with precedence constraints," *Eur. J. Oper. Res.*, vol. 282, no. 2, pp. 515–529, Apr. 2020.

[20] 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, Apr. 2019.

[21] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles," *ACM Trans. Sensor Netw.*, vol. 12, no. 2, pp. 1–26, May 2016.

[22] W. Liang, Z. Xu, W. Xu, J. Shi, G. Mao, and S. K. Das, "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, Oct. 2017.

[23] L. Lin and M. A. Goodrich, "Hierarchical heuristic search using a Gaussian mixture model for UAV coverage planning," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2532–2544, Dec. 2014.

[24] C. Lin, C. Guo, H. Dai, L. Wang, and G. Wu, "Near optimal charging scheduling for 3-D wireless rechargeable sensor networks with energy constraints," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 624–633.

[25] C. Lin, C. Guo, J. Deng, and G. Wu, "3DCS: A 3-D dynamic collaborative scheduling scheme for wireless rechargeable sensor networks with heterogeneous chargers," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 320–331.

[26] T. Liu, B. Wu, H. Wu, and J. Peng, "Low-cost collaborative mobile charging for large-scale wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2213–2227, Aug. 2017.

[27] V. Mersheeva and G. Friedrich, "Multi-UAV monitoring with priorities and limited energy resources," in *Proc. 25th Int. Conf. Int. Conf. Automated Planning Scheduling*, 2015, pp. 327–356.

[28] 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, Dec. 1978.

[29] C. Orlis, N. Bianchessi, R. Roberti, and W. Dullaert, "The team orienteering problem with overlaps: An application in cash logistics," *Transp. Sci.*, pp. 470–487, Mar. 2020.

[30] A. Paul, D. Freund, A. Ferber, D. Shmoys, and D. Williamson, "Prize-collecting TSP with a budget constraint," in *Proc. 25th Annu. Eur. Symp. Algorithms (ESA)*. Wadern, Germany: Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017, p. 62.

[31] T. Liu, B. Wu, S. Zhang, J. Peng, and W. Xu, "An effective multi-node charging scheme for wireless rechargeable sensor networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 2026–2035.

[32] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: A survey," *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 1–10, Feb. 2011.

[33] T. Vidal, N. Maculan, L. S. Ochi, and P. H. Vaz Penna, "Large neighborhoods with implicit customer selection for vehicle routing problems with profits," *Transp. Sci.*, vol. 50, no. 2, pp. 720–734, May 2016.

[34] C. Wang, J. Li, F. Ye, and Y. Yang, "A mobile data gathering framework for wireless rechargeable sensor networks with vehicle movement costs and capacity constraints," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2411–2427, Aug. 2016.

[35] W. Xu, W. Liang, J. Peng, Y. Liu, and Y. Wang, "Maximizing charging satisfaction of smartphone users via wireless energy transfer," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 990–1004, Apr. 2017.

[36] W. Xu, W. Liang, X. Jia, Z. Xu, Z. Li, and Y. Liu, "Maximizing sensor lifetime with the minimal service cost of a mobile charger in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 11, pp. 2564–2577, Nov. 2018.

[37] W. Xu, W. Liang, H. Kan, Y. Xu, and X. Zhang, "Minimizing the longest charge delay of multiple mobile chargers for wireless rechargeable sensor networks by charging multiple sensors simultaneously," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2019, pp. 881–890.

[38] W. Xu, W. Liang, X. Jia, H. Kan, Y. Xu, and X. Zhang, "Minimizing the maximum charging delay of multiple mobile chargers under the multi-node energy charging scheme," *IEEE Trans. Mobile Comput.*, early access, Feb. 14, 2020, doi: 10.1109/TMC.2020.2973979.

[39] W. Xu *et al.*, "Approximation algorithms for the team orienteering problem," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Jul. 2020, pp. 1389–1398.

[40] V. F. Yu, P. Jewpanya, S.-W. Lin, and A. A. N. P. Redi, "Team orienteering problem with time windows and time-dependent scores," *Comput. Ind. Eng.*, vol. 127, pp. 213–224, Jan. 2019.

[41] S. Zhang, J. Wu, and S. Lu, "Collaborative mobile charging," *IEEE Trans. Comput.*, vol. 64, no. 3, pp. 654–667, Mar. 2015.

Wenzheng Xu (Member, IEEE) received the B.Sc., M.E., and Ph.D. degrees in computer science from Sun Yat-sen University, Guangzhou, China, in 2008, 2010, and 2015, respectively. He was a Visitor at The Australian National University and The Chinese University of Hong Kong. He is currently an Associate Professor with Sichuan University. His research interests include wireless ad hoc and sensor networks, UAV networks, mobile computing, approximation algorithms, combinatorial optimization, online social networks, and graph theory.

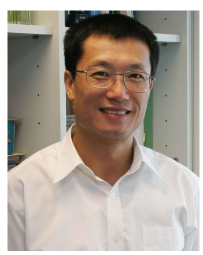
Weifa Liang (Senior Member, IEEE) received the B.Sc. degree from Wuhan University, China, in 1984, the M.E. degree from the University of Science and Technology of China in 1989, and the Ph.D. degree from The Australian National University in 1998, all in computer science. He is currently a Full Professor with The Australian National University. His research interests include design and analysis of energy-efficient routing protocols for wireless ad hoc and sensor networks, mobile edge computing and cloud computing, software-defined networking, online social networks, design and analysis of parallel and distributed algorithms, approximation algorithms, combinatorial optimization, and graph theory.

Zichuan Xu (Member, IEEE) received the B.Sc. and M.E. degrees from the Dalian University of Technology, China, in 2008 and 2011, respectively, and the Ph.D. degree from The Australian National University in 2016, all in computer science. He was a Research Associate with University College London. He is currently an Associate Professor with the School of Software, Dalian University of Technology. His research interests include cloud computing, software-defined networking, wireless sensor networks, algorithmic game theory, and optimization problems.

Jian Peng received the B.A. and Ph.D. degrees from the University of Electronic Science and Technology of China (UESTC) in 1992 and 2004, respectively. He is currently a Professor with the College of Computer Science, Sichuan University. His recent research interests include wireless sensor networks, big data, and cloud computing.

Dezhong Peng (Member, IEEE) received the B.Sc. degree in applied mathematics and the M.Sc. and Ph.D. degrees in computer software and theory from the University of Electronic Science and Technology of China, Chengdu, China, in 1998, 2001, and 2006, respectively. From 2001 to 2007, he was an Assistant Lecturer and a Lecturer with the University of Electronic Science and Technology of China. He was a Post-Doctoral Research Fellow with the School of Engineering, Deakin University, Burwood, VIC, Australia, from 2007 to 2009. He is currently a Professor with the Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu. His current research interests include artificial intelligence and big data.

Tang Liu received the B.S. degree in computer science from the University of Electronic and Science of China in 2003 and the M.S. and Ph.D. degrees in computer science from Sichuan University in 2009 and 2015, respectively. Since 2003, he has been with the College of Computer Science, Sichuan Normal University, where he is currently a Professor. From 2015 to 2016, he was a Visiting Scholar with the University of Louisiana at Lafayette. His research interests include wireless charging and wireless sensor networks.



Xiaohua Jia (Fellow, IEEE) received the B.Sc. and M.Eng. degrees from the University of Science and Technology of China in 1984 and 1987, respectively, and the D.Sc. degree in information science from The University of Tokyo in 1991. He is currently a Chair Professor with the Department of Computer Science, City University of Hong Kong. His research interests include cloud computing and distributed systems, computer networks, wireless sensor networks, and mobile wireless networks. He is the General Chair of ACM MobiHoc 2008, the TPC Co-Chair of IEEE MASS 2009, the Area-Chair of IEEE INFOCOM 2010, the TPC Co-Chair of IEEE GLOBECOM 2010 and the Ad Hoc and Sensor Networking Symposium, and the Panel Co-Chair of IEEE INFOCOM 2011. He was an Editor of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS (from 2006 to 2009), *World Wide Web* Journal, *Wireless Networks*, *Journal of Combinatorial Optimization*, and so on.

Sajal K. Das (Fellow, IEEE) is currently the Chair of the Computer Science Department and the Daniel St. Clair Endowed Chair of the Missouri University of Science and Technology, USA. His current research interests include theory and practice of wireless sensor networks, big data, cyber-physical systems, smart healthcare, distributed and cloud computing, security and privacy, biological and social networks, applied graph theory, and game theory. He directed numerous funded projects in these areas totaling over \$15M and published extensively with more than 600 research articles in high-quality journals and refereed conference proceedings. He serves as the Founding Editor-in-Chief of the *Pervasive and Mobile Computing* Journal and an Associate Editor for the IEEE TRANSACTIONS ON MOBILE COMPUTING, the *ACM Transactions on Sensor Networks*, and so on. He is a Co-Founder of the IEEE PerCom, IEEE WoWMoM, and ICDCN conferences, and served on numerous conference committees as the general chair, the program chair, or a program committee member.