Approximation Algorithms for the Team Orienteering Problem

Wenzheng Xu[a], Zichuan Xu[b], Jian Peng[a], Weifa Liang[c], Tang Liu*[d], Xiaohua Jia[e], and Sajal K. Das[f]

[a]College of Computer Science, Sichuan University, Chengdu, 510006, P. R. China

[b]School of Software, Dalian University of Technology, Dalian, 116024, P. R. China

[c]Research School of Computer Science, The Australian National University, Canberra, ACT 0200, Australia

[d]College of Computer Science, Sichuan Normal University, Chengdu, Sichuan, 610068, P. R. China

[e]Department of Computer Science, City University of Hong Kong, Hong Kong, P. R. China

[f]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409 USA

Emails: wenzheng.xu3@gmail.com, z.xu@dlut.edu.cn, jianpeng@scu.edu.cn, wliang@cs.anu.edu.au, liutang@sicnu.edu.cn, csjia@cityu.edu.hk, sdas@mst.edu

*Corresponding author: Tang Liu

***Abstract*—In this paper we study a team orienteering problem, which is to find service paths for multiple 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 IoT 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 paper, we first formulate the team orienteering problem, where different vehicles are different types, 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 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 no less than 0.32 when $\epsilon = 0.5$. In addition, for a special team orienteering problem with the same type of vehicles and the profits of serving a node once and multiple times being the same, we devise an improved approximation algorithm. Finally, we evaluate the proposed algorithms with simulation experiments, and the results of which are very promising. Precisely, the profit sums delivered by the proposed algorithms are approximately 12.5% to 17.5% higher than those by existing algorithms.**

***Index Terms*—Multiple vehicle scheduling; team orienteering problem; approximation algorithms; submodular function.**

I. Introduction

In this paper we consider a generalized orienteering problem – the *team orienteering problem*, which has wide applications in the areas of Internet of Things (IoTs) and smart cities [14], [24], [27]. For example, given multiple mobile chargers, consider the problem of how to dispatch them to recharge energy-critical IoT sensors in a monitoring region by scheduling their charging tours [8], [17], [19], [20], [21], [28], [29], [30], [31], [33], [34]. Fig. 1 illustrates the employment of two mobile chargers to recharge sensors in a wireless sensor network. When many energy-critical sensors are 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 them. Therefore, a fundamental problem is to schedule the chargers to recharge a portion of energy-critical sensors such that the charging profit sum of the sensors is maximized, subject to the energy capacity of each mobile charger. It is understood that more charging profit can be collected by charging a sensor with low residual energy than by charging a sensor with high residual energy [17], [19].

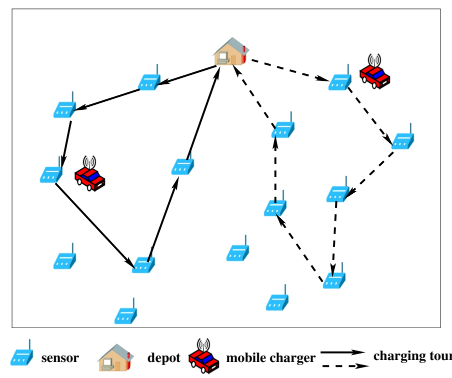


Fig. 1. An example of the team orienteering problem for dispatching two mobile chargers to recharge energy-critical sensors in a sensor network.

Other applications of the team orienteering problem include scheduling multiple Unmanned Aerial Vehicles (UAVs) in disaster areas to monitor Points of Interest (POIs), such as malls, schools, office buildings, etc. [9], [10], [15], [18], [22], [26], and dispatching a fleet of autonomous vehicles to deliver goods to different households in the scenario of smart cities.

As mentioned, the team orienteering problem is a generalization of the traditional orienteering problem [14], [27]. 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 $v_i \in V$. Then 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 fact 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. Blum *et al.* [3] proposed the first constant approximation algorithm for this problem with a ratio of $\frac{1}{4}$. Bansal *et al.* [2] shortly improved the approximation ratio to $\frac{1}{3}$. Recently, Chekuri *et al.* [6] further improved the ratio to $\frac{1}{2+\epsilon}$ when the starting node s and ending node t may be different, where ϵ is a given constant with $0 < \epsilon \leq 1$. Moreover, Paul *et al.* [24] devised a $\frac{1}{2}$-approximation algorithm when the nodes s and t are co-located (i.e., $s = t$).

Although the orienteering problem has been well studied, many applications need to find paths for multiple vehicles rather than just one. 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. However, to the best of our knowledge there are no any performance-guaranteed algorithms for such multi-vehicle case.

In this paper, we study the *team orienteering problem*, which is to find service paths for $K \geq 1$ 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 constraint on each vehicle. Furthermore, we consider several *generalized scenarios* of the team orienteering problem in practical applications, such as:

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

(ii) Different types of vehicles (rather than the same type) may also be considered. Different types of vehicles may have different travel and service costs, as well as different cost budgets. For example, the maximum flying duration of a 'DJI Phantom 4 Pro' UAV is approximately 30 minutes, while a 'DJI Inspire 1' UAV can fly for only approximately 18 minutes with its battery capacity.

(iii) Each node may be served by multiple vehicles (rather than by only one) and a nondecreasing submodular function can be adopted to model the profit of serving a node. In other words, the more vehicles serve a node, the less marginal profit is. For example, consider the deployment of 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 [19]. However, photos taken by different UAVs may have some 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 paper lies in formulating a novel problem, namely the *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. Our major contributions are summarized as follows.

- We are the first to study the team orienteering problem, where different types of vehicles have different travel costs, node service costs, and cost budgets. 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 that node.
- We propose a novel $(1-(1/e)^{\frac{1}{2+\epsilon}})$-approximation algorithm for the team orienteering 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 no less than 0.32 when $\epsilon = 0.5$.
- We devise an improved approximation algorithm for a special case of the team orienteering problem where all vehicles are the same type and the profits that a node is served once and multiple times are the same.
- We evaluate the proposed algorithms with simulations. Experiment results show that the profit sums delivered by the proposed algorithms are approximately 12.5% to 17.5% higher than those by existing algorithms.

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

II. 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 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 $K \geq 1$ 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 $1 \leq k \leq K$.

We assume that the K vehicles are different types and thus have different travel and service costs. Denote by $c_k(v_i, v_{i+1})$ the cost of vehicle k for traveling between nodes v_i and v_{i+1}, and $h_k(v_i)$ the service cost of vehicle k at node v_i. Assume that $h_k(s) = h_k(t) = 0$. The cost $w(P_k)$ of path P_k for vehicle k then is

$$w(P_k) = \sum_{i=1}^{q_k} h_k(v_i) + \sum_{i=0}^{q_k} c_k(v_i, v_{i+1}), \tag{1}$$

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

Notice that different types of vehicles have different cost budgets, where the cost budget of a vehicle is either its energy capacity or its maximum operation duration per tour. Denote by B_k the cost budget of vehicle k with $1 \le k \le K$. The cost $w(P_k)$ of path P_k for vehicle k thus must be no greater than its cost budget B_k, i.e., $w(P_k) \le B_k$.

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 $0 \le n_i \le K$. 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: $0 \le u_i(x) \le u_i(y)$ if $0 \le x \le y$, where x and y are two integers; and (iii) the submodularity property: for any nonnegative integer Δ, $u_i(x+\Delta) - u_i(x) \ge u_i(y+\Delta) - u_i(y)$ if $0 \le x \le y$. 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 two examples. One example is that $u_i(n_i) = 1$ if $n_i \ge 1$; otherwise, $u_i(n_i) = 0$. 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 $u_i(n_i) = \log_2(n_i+1)$, which implies that the more vehicles serve a node v_i, the less the marginal gain is obtained from serving it. 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 zero, i.e., $u_s(n_s) = u_t(n_t) = 0$ with $0 \le n_s, n_t \le K$.

The total profit received from serving the nodes in the K paths $P_1, P_2, \ldots, P_K$ then is $\sum_{v_i \in \cup_{k=1}^{K} P_k} u_i(n_i)$, 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 team orienteering problem* 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., $\sum_{v_i \in \cup_{k=1}^{K} P_k} u_i(n_i)$, is maximized, subject to that the cost $w(P_k)$ of each path P_k for vehicle k is no greater than its cost budget B_k, i.e., $w(P_k) \le B_k$ with $1 \le k \le K$. That is,

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

subject to

$$w(P_k) \le B_k, \quad 1 \le k \le K. \tag{3}$$

The team orienteering problem is NP-hard, since the orienteering problem is its special case when $K = 1$.

III. Approximation Algorithm

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

The basic idea of the proposed algorithm is that it proceeds iteratively. Within each iteration, one path with the maximum marginal gain is found for one vehicle in an auxiliary graph. Thus, there are K iterations. The detailed algorithmic description is as follows.

A. Algorithm

Given graph $G = (V \cup \{s,t\}, E)$, the algorithm first constructs K auxiliary graphs $G_1, G_2, \ldots, G_K$, where graph G_k is for finding the path P_k of vehicle k, $G_k = (V \cup \{s,t\}, E; w_k : E \mapsto R^{\ge 0})$, and the weight $w_k(v_i, v_j)$ of each edge (v_i, v_j) in G_k is

$$w_k(v_i, v_j) = c_k(v_i, v_j) + \frac{h_k(v_i) + h_k(v_j)}{2}, \tag{4}$$

$c_k(v_i, v_j)$ is the traveling cost between nodes v_i and v_j for vehicle k, and $h_k(v_i)$ and $h_k(v_j)$ are the service costs of vehicle k at nodes v_i and v_j, respectively. For any $s-t$ path P_k in G_k, denote by $w_k(P_k)$ the weighted sum of the edges in P_k, i.e., $w_k(P_k) = \sum_{(v_i,v_j) \in P_k} w_k(v_i, v_j)$.

There are two interesting relationships between graphs G and G_k, 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 is equal to the weighted sum $w_k(P_k)$ of the edges in P_k of G_k.
(ii) The edge weights in G_k satisfy the triangle inequality.

The proposed algorithm then finds K paths for the K vehicles. Assume that it has found K' paths $P_{l_1}, P_{l_2}, \ldots, P_{l_{K'}}$, where P_{l_j} is the path for vehicle l_j, $1 \le l_j \le K$ and $1 \le j \le K'$. Also, assume that each node v_i in V has been served n_i times with $0 \le n_i \le K'$. Initially, $K' = 0$ and $n_i = 0$ for each $v_i \in V$. The algorithm now finds the $(K'+1)$th path $P_{l_{K'+1}}$ as follows.

Let $\mathcal{K} = \{1, 2, \ldots, K\}$ be the set of the K vehicles. Also, let $\mathcal{L} = \{l_1, l_2, \ldots, l_{K'}\}$ the set of the vehicles that have been assigned paths already. For each vehicle k that has not been assigned a path (i.e., $k \in \mathcal{K} \setminus \mathcal{L}$), it finds an approximate $s-t$ path P_k in G_k under the cost budget B_k constraint, by applying an approximation algorithm for the orienteering problem, where the profit of serving a node v_i in G_k is set to $u(v_i, K'+1) = u_i(n_i+1) - u_i(n_i)$, and v_i has already been served n_i times in the previous K' paths. Denote by $u(P_k, K'+1)$ the profit sum of the nodes in path P_k, i.e., $u(P_k, K'+1) = \sum_{v_i \in P_k} u(v_i, K'+1)$.

The $(K'+1)$th path $P_{l_{K'+1}}$ for vehicle $l_{K'+1}$ then is the path with the maximum profit, i.e.,

$$l_{K'+1} = \arg\max_{k \in \mathcal{K} \setminus \mathcal{L}} \{u(P_k, K'+1)\}. \tag{5}$$

After finding the $(K'+1)$the path, the algorithm increases the value of K' by one, updates the number of times n_i that v_i

Algorithm 1 Approximation algorithm for the team orienteering problem (`approAlg`)

Input: $G = (V \cap \{s,t\}, E)$, K different types of vehicles with cost budgets $B_1, B_2, \ldots, B_K$, travel cost $c_k : E \mapsto R^{\geq 0}$, service cost $h_k : V \mapsto R^{\geq 0}$, and profit $u_i : Z^{\geq 0} \mapsto R^{\geq 0}$ for each $v_i \in V$.

Output: 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 K auxiliary graphs $G_1, G_2, \ldots, G_K$ from G, where $G_k = (V \cap \{s,t\}, E)$, $w_k : E \mapsto R^{\geq 0}$, and $w_k(v_i, v_j) = c_k(v_i, v_j) + \frac{h_k(v_i)+h_k(v_j)}{2}$ for each edge (v_i, v_j) in G_k;
Let $\mathcal{K} = \{1, 2, \ldots, K\}$; /* the set of K vehicles */
Let $\mathcal{L} = \emptyset$; /* the set of vehicles that have been found paths for them */
Let $\mathcal{P} = \emptyset$; /* 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**
  For each vehicle k in $\mathcal{K} \setminus \mathcal{L}$, find an approximate $s-t$ path P_k in G_k with cost budget B_k, by applying an approximation algorithm for the orienteering problem, where the profit for serving each node v_i in G_k is set as $u(v_i, K') = u_i(n_i + 1) - u_i(n_i)$;
  Choose a path $P_{l_{K'+1}}$ for vehicle $l_{K'+1}$ with the maximum profit, i.e., $l_{K'+1} = \arg\max_{k \in \mathcal{K} \setminus \mathcal{L}}\{u(P_k, K')\}$;
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{P_{l_{K'+1}}\}$;
  For each v_i in path $P_{l_{K'+1}}$, increase its number of served times n_i by one;
  $\mathcal{L} \leftarrow \mathcal{L} \cup \{l_{K'+1}\}$;
  $K' \leftarrow K' + 1$;
end while
return K paths in $\mathcal{P}$.

is served, and the set $\mathcal{L}$. The algorithm continues until the K paths are found.

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

B. Algorithm analysis

For the sake of convenience, we assume that the K'th path $P_{l_{K'}}$ delivered by `Algorithm` 1 is for vehicle K', i.e., $l_{K'} = K'$ for each K' with $1 \leq K' \leq K$.

Theorem 1: Given a graph $G = (V \cup \{s,t\}, E)$, K vehicles with vehicle k having a cost budget B_k, the travel cost function $c_k : E \mapsto R^{\geq 0}$, service cost function $h_k : V \mapsto R^{\geq 0}$, and the profit function $u_i : Z^{\geq 0} \mapsto R^{\geq 0}$ for each node v_i in V, there is an approximation algorithm, `Algorithm` 1, for the team orienteering 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 $P_1^*, P_2^*, \ldots, P_K^*$ form an optimal solution to the team orienteering 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 the 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.,

$$\hat{P}_{k+1} = \arg\max_{P \in \mathcal{P}}\{u(\mathcal{A}_k \cup \{P\}) - u(\mathcal{A}_k)\}, \tag{6}$$

where $\mathcal{P}$ is the set of all feasible paths for the $(k+1)$th, $(k+2)$th, …, and Kth vehicles.

Denote by $P_{k,j}^*$ the optimal path for the orienteering problem in G_j with $k+1 \leq j \leq K$, where the profit $u(v_i, k)$ for serving each node v_i in G_j is $u(v_i, k) = u_i(n_i^{\mathcal{A}_k} + 1) - u_i(n_i^{\mathcal{A}_k})$, and $n_i^{\mathcal{A}_k}$ is the number of times that v_i is served by the paths in $\mathcal{A}_k$. It can be seen that $\hat{P}_{k+1}$ is the path with the maximum profit among the $K-k$ paths $P_{k,k+1}^*, P_{k,k+2}^*, \ldots, P_{k,K}^*$, i.e.,

$$\hat{P}_{k+1} = \arg\max_{k+1 \leq j \leq K}\{u(P_{k,j}^*, k)\}. \tag{7}$$

For the sake of convenience, assume that $\hat{P}_{k+1} = P_{k,j^*}^*$, where $k+1 \leq j^* \leq K$.

Recall that, an α-approximate path P_j' for the orienteering problem in graph G_j can be delivered by `Algorithm` 1, i.e., $u(P_j') \geq \alpha \cdot u(P_{k,j}^*)$ with $k+1 \leq j \leq K$. On the other hand, recall that P_{k+1} is the path with the maximum profit among the $K-k$ paths $P_{k+1}', P_{k+2}', \ldots, P_K'$ found by `Algorithm` 1. We thus have

$$\begin{aligned}
& u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k) \\
= \;& u(P_{k+1}) \\
= \;& \arg\max_{k+1 \leq j \leq K}\{u(P_j')\} \\
\geq \;& u(P_{j^*}') \\
\geq \;& \alpha \cdot u(P_{k,j^*}^*) \\
= \;& \alpha \cdot u(\hat{P}_{k+1}), \\
= \;& \alpha \cdot (u(\mathcal{A}_k \cup \{\hat{P}_{k+1}\}) - u(\mathcal{A}_k)),
\end{aligned} \tag{8}$$

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

$$u(\mathcal{A}_k \cup \mathcal{A}_j^*) \;\leq\; \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*) \tag{9}$$

as follows.

$$\begin{aligned}
& u(\mathcal{A}_k \cup \mathcal{A}_j^*) \\
= \;& u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^* \cup \{P_j^*\}), \text{ by the definition of } \mathcal{A}_j^* \\
= \;& u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^* \cup \{P_j^*\}) - u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*) \\
\leq \;& u(\mathcal{A}_k \cup \{P_j^*\}) - u(\mathcal{A}_k) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*), \\
& \quad \text{due to the submodularity of } u(.) \text{ and } \mathcal{A}_k \subseteq \mathcal{A}_k \cup \mathcal{A}_{j-1}^* \\
\leq \;& u(\mathcal{A}_k \cup \{\hat{P}_{k+1}\}) - u(\mathcal{A}_k) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*), \\
& \text{since } \hat{P}_{k+1} \text{ is an optimal path with respect to } \mathcal{A}_k \\
\leq \;& \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}_{j-1}^*), \text{due to Eq. (8)}
\end{aligned}$$

We then have

$$\begin{aligned}
u(\mathcal{A}^*) &= u(\mathcal{A}^*_K), \text{ as } \mathcal{A}^* = \mathcal{A}^*_K \\
&\le u(\mathcal{A}_k \cup \mathcal{A}^*_K), \text{ as function } u(.) \text{ is nondecreasing} \\
&\le \frac{1}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}^*_{K-1}), \\
&\qquad \text{due to Ineq. (9) where } j = K, \\
&\le \frac{2}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}^*_{K-2}), \\
&\qquad \text{due to Ineq. (9) where } j = K-1, \\
&\vdots \\
&\le \frac{K}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k \cup \mathcal{A}^*_{K-K}), \\
&= \frac{K}{\alpha}(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k), \\
&\qquad \text{where } \mathcal{A}^*_{K-K} = \mathcal{A}^*_0 = \emptyset \\
&= \beta K(u(\mathcal{A}_{k+1}) - u(\mathcal{A}_k)) + u(\mathcal{A}_k), \text{ let } \beta = \frac{1}{\alpha}. \quad (10)
\end{aligned}$$

Re-arranging Ineq. (10), we have

$$\begin{aligned}
u(\mathcal{A}_{k+1}) &\ge \frac{\beta K - 1}{\beta K} u(\mathcal{A}_k) + \frac{u(\mathcal{A}^*)}{\beta K} \\
&= a \cdot u(\mathcal{A}_k) + b, \quad (11)
\end{aligned}$$

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

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

$$\begin{aligned}
u(\mathcal{A}_K) &\ge a \cdot u(\mathcal{A}_{K-1}) + b, \text{ by Eq. (11)} \\
&\ge a(a \cdot u(\mathcal{A}_{K-2}) + b) + b \\
&= a^2 \cdot u(\mathcal{A}_{K-2}) + ab + b \\
&\vdots \\
&\ge a^K \cdot u(\mathcal{A}_{K-K}) + b \sum_{k=0}^{K-1} a^k \\
&= b \sum_{k=0}^{K-1} a^k, \text{ as } \mathcal{A}_{K-K} = \emptyset \text{ and } u(\emptyset) = 0 \\
&= b \frac{1 - a^K}{1 - a} \\
&= \frac{u(\mathcal{A}^*)}{\beta K} \frac{1 - (\frac{\beta K - 1}{\beta K})^K}{1 - \frac{\beta K - 1}{\beta K}}, \\
&= u(\mathcal{A}^*)(1 - (1 - \frac{1}{\beta K})^K) \\
&= u(\mathcal{A}^*)(1 - ((1 - \frac{1}{\beta K})^{\beta K})^{\frac{1}{\beta}}) \\
&\ge u(\mathcal{A}^*)(1 - (1/e)^{\frac{1}{\beta}}), \text{ as } (1 - \frac{1}{\beta K})^{\beta K} \le \frac{1}{e} \\
&= (1 - (1/e)^{\alpha}) u(\mathcal{A}^*), \text{ as } \alpha = \frac{1}{\beta}. \quad (12)
\end{aligned}$$

The theorem then follows. ■

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

Proof: Following Chekuri *et al.* [6], 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 $0 < \epsilon \le 1$. The lemma immediately follows, due to Theorem 1. When $\epsilon = 0.5$, the approximation ratio is $1 - (1/e)^{0.4} \ge 0.32$. ■

In practice, 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 vehicles and we can obtain a better approximation ratio as follows.

Corollary 2: There is a $(1 - 1/\sqrt{e})$-approximation algorithm for the team orienteering problem, when the starting node s and ending node t are co-located, where e is the base of the natural logarithm.

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.* [24], the approximation ratio of `Algorithm 1` then is $1 - 1/\sqrt{e}$, which is no less than 0.39. ■

IV. Approximation Algorithm with the Same Type of Vehicles

In this section, we consider a special case of the team orienteering problem, where the K vehicles are the same type, the source node s and the destination node t are co-located (i.e., $s = t$), and the profits $u_i(n_i)$ that each node v_i is served once and multiple times are 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.

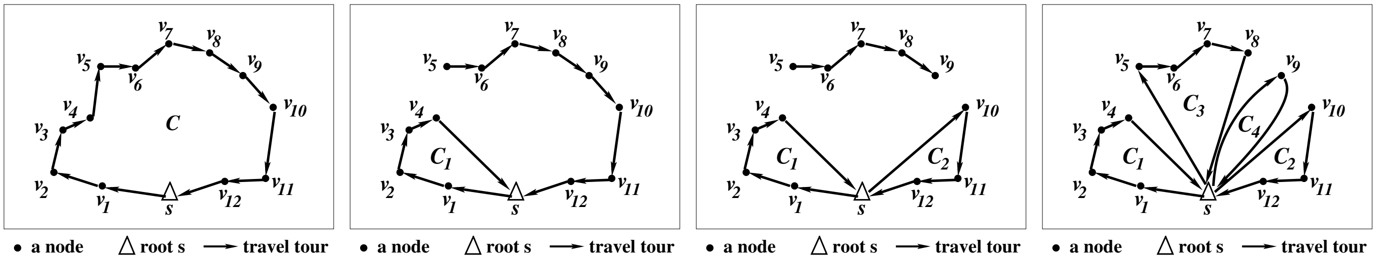
Formally, given a graph $G = (V \cup \{s\}, E)$, K vehicles with the same type, a cost budget B of each vehicle, a travel cost function $c : E \mapsto R^{\ge 0}$, a node service cost function $h : V \mapsto R^{\ge 0}$, and a profit function $u : V \mapsto R^{\ge 0}$, *the special team orienteering problem* 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, $\sum_{v_i \in \cup_{k=1}^K C_k} u(v_i)$, is maximized, subject to that the cost $w(C_k)$ of each tour C_k is no greater than B, where $1 \le k \le K$.

Although the proposed greedy algorithm in the previous section is applicable for this special team orienteering problem with an approximation ratio of 0.39 by Corollary 2, we here devise an approximation algorithm for this special problem, and we will show that the algorithm is able to find a better solution than that by the 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 $K' \ge K$. 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



(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. 2. Illustration of the approximation algorithm for the special team orienteering problem, where $K = 3$.

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 $G' = (V \cup \{s\}, E; w : E \mapsto R^{\geq 0})$ 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.

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 from [24], 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 in C, e.g., see Fig. 2(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, which will be 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. 2(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. 2(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. 2(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. 2(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.

Algorithm 2 Approximation algorithm for the special team orienteering problem (`approAlgSpecial`)

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

Output: 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.

1: Construct an auxiliary graph $G' = (V \cap \{s\}, E; w : E \mapsto R^{\geq 0})$ 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';
2: Find a $\frac{1}{2}$-approximate tour C for the orienteering problem in G' with cost budget KB, by invoking an algorithm from [24];
3: 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;
4: Let $\mathcal{C}$ be the set of the K tours with the maximum profits among the K' tours;
5: **return** the K tours in $\mathcal{C}$.

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., $u(C_k) = \sum_{v_i \in C_k} u(v_i)$, where $1 \leq k \leq K'$. For the sake of convenience, we assume that $u(C_1) \geq u(C_2) \geq \cdots \geq u(C_{K'})$. 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 special team orienteering problem if $K \leq K'$. 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 special team orienteering problem is presented in `Algorithm` 2.

B. Algorithm analysis

We assume that $K \geq 2$. Otherwise (i.e., $K = 1$), the special team orienteering problem degenerates to the orienteering problem. In the following, we first obtain a nontrivial upper bound on the special 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., $OPT = \sum_{v_i \in \cup_{k=1}^{K} C_k^*} u(v_i)$. Meanwhile, denote by C_L^* the optimal solution to the orienteering problem in G' with cost budget KB.

We start with the following important lemma.

Lemma 1: The optimal value OPT of the special 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., $OPT \leq u(C^*_L)$.

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. Then, it can be seen that tour C is a feasible solution to the orienteering problem in G' with cost budget KB. Since C^*_L is the optimal solution, we have

$$OPT = \sum_{v_i \in \cup_{k=1}^{K} V(C^*_k)} u(v_i) \; = \; u(C) \; \leq \; u(C^*_L). \tag{13}$$

The lemma then follows. ■

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

Lemma 2: `Algorithm` 2 delivers a $\frac{1}{2}$-approximate solution to the special team orienteering problem in G if $K' \leq K$.

Proof: Since $K' \leq K$, the solution delivered by `Algorithm` 2 consists of the K' split tours $C_1, C_2, \ldots, C_{K'}$. We then have

$$\begin{aligned}
&\sum_{v_i \in \cup_{k=1}^{K'} C_k} u(v_i) \; = \; u(C) \\
\geq \;& \frac{1}{2} \cdot u(C^*_L), \text{ as } C \text{ is a } \frac{1}{2}\text{-approximate solution} \\
\geq \;& \frac{1}{2} \cdot OPT, \text{ by Lemma 1.}
\end{aligned} \tag{14}$$

■

We consider case (ii) that $K' > K$ as follows.

Lemma 3: `Algorithm` 2 delivers an α-approximate solution to the special team orienteering problem in G, if $K' > K \geq 2$, 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 $\Delta \leq \frac{B}{2}$.

Proof: 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

$$\begin{aligned}
K' \;&\leq\; 2 + \lceil \frac{(K-2)B + 2\Delta}{B - 2\Delta} \rceil \\
&=\; \lceil \frac{(K-2)B + 2\Delta}{B - 2\Delta} + 2 \rceil \\
&=\; \lceil \frac{KB - 2\Delta}{B - 2\Delta} \rceil \;\leq\; \lceil \frac{KB}{B - 2\Delta} \rceil.
\end{aligned} \tag{15}$$

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

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

$$\begin{aligned}
\frac{u(\mathcal{C})}{OPT} \;&\geq\; \frac{\frac{K}{K'} \cdot \sum_{v_i \in \cup_{k=1}^{K'} V(C_k)}}{OPT} \\
&\geq\; \frac{\frac{K}{K'} \cdot \frac{1}{2}OPT}{OPT}, \text{ by Ineq. (14)} \\
&=\; \frac{1}{2} \cdot \frac{K}{K'} \\
&\geq\; \frac{1}{2} \cdot \frac{K}{\lceil \frac{KB}{B-2\Delta} \rceil}, \text{ by Ineq. (15)}
\end{aligned} \tag{16}$$

Remark: We compare the approximation ratio $\frac{K}{2\lceil \frac{KB}{B-2\Delta} \rceil}$ of `Algorithm` 2 against the ratio $(1 - (1/e)^{\frac{1}{2}})$ of `Algorithm` 1. On one hand, the value of $(1 - (1/e)^{\frac{1}{2}})$ 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 $\frac{K}{2\lceil \frac{KB}{B-2\Delta} \rceil} \approx \frac{K}{2\frac{KB}{B-2\Delta}} = \frac{B-2\Delta}{2B} \geq 0.4 \geq 1 - (1/e)^{\frac{1}{2}}$, if $B \geq 10\Delta$. 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. ■

V. Performance Evaluation

A. Environmental setting

We consider an application of the team orienteering problem for scheduling multiple mobile chargers to charge sensors in a sensor network. There are from 50 to 200 sensors deployed in a 1000 $m \times 1000$ m square area randomly [20]. The battery capacity of each sensor v_i is 10.8 kJ [17]. 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 10. Since the K mobile chargers are different types, we assume that the energy capacity B_k of charger k is randomly chosen from an interval $[1,000 \; kJ, \; 1,500 \; kJ]$ [16]. Also, the traveling energy consumption of charger k between any two nodes v_i and v_j

is $c_k(v_i, v_j) = \eta_k \cdot d_{ij}$, where η_k is its energy consumption on traveling per unit length that is randomly chosen from an interval $[0.5\ kJ/m,\ 0.6\ kJ/m]$ [16], 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 charger k is $h_k(v_i) = \frac{10.8\ kJ - re_i}{\rho_k}$, where $10.8\ kJ - re_i$ is the amount of energy charged to v_i, ρ_k is the energy charging efficiency of charger k, and ρ_k is randomly chosen from an interval $[0.9,\ 0.95]$.

We here consider two algorithms for the benchmark purpose against the proposed algorithms.

(1) Algorithm `partitionAlg` first sorts all sensors in anticlockwise order with centering at depot s. Assume that $v_1, v_2, \ldots, v_n$ is the order of sorted sensors. The algorithm then partitions the sensors into K disjoint sets $V_1, V_2, \ldots, V_K$, by the energy capacities of the K chargers. That is, sensors $v_1, v_2, \ldots, v_{n_1}$ are in set V_1, where $n_1 = \lceil n \cdot \frac{B_1}{\sum_{k=1}^{K} B_k} \rceil$; sensors $v_{n_1+1}, v_{n_1+2}, \ldots, v_{n_2}$ are in set V_2, where $n_2 = \lceil n \cdot \frac{B_1+B_2}{\sum_{k=1}^{K} B_k} \rceil$; $\ldots$; sensors $v_{n_{K-1}+1}, v_{n_{K-1}+2}, \ldots, v_{n_K}$ are in set V_K, where $n_K = \lceil n \cdot \frac{\sum_{k=1}^{K} B_k}{\sum_{k=1}^{K} B_k} \rceil = n$. It can be seen that the larger the energy capacity B_k of a charger k is, the more sensors in set V_k are. After partitioning the sensors, the algorithm finds a charging tour C_k of charger k for charging the sensors in each set V_k by applying the approximation algorithm for the orienteering problem [24].

(2) Algorithm `forestAlg` finds the K charging tours by starting with a forest with K trivial trees and each tree T_k consists of only depot s initially [1]. Assume that some sensors have been inserted to the forest already. For each sensor v_i that is not in the forest, the algorithm calculates both the profit $u_i(v_i)$ and 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 closed tour C_k consisting of nodes in T_k and the cost of the tour C'_k consisting of both nodes in T_k and v_i, and the cost of C'_k is no larger than B_k, i.e., $\delta_i = \min_{k=1, w(C'_k) \le B_k}^{K} \{w(C'_k) - w(C_k)\}$. The algorithm then inserts a sensor v_j to the forest with the maximum ratio of the profit $u_i(v_i)$ to the increased cost δ_i of serving a sensor v_i , i.e., $v_j = \arg\max_{v_i \in V \setminus \cup_{k=1}^{K} T_k} \{\frac{u_i(v_i)}{\delta_i}\}$. This procedure of the forest growth continues until either the insertion of any sensor will violate the cost budgets of the K vehicles or all sensors have been included in the forest.

B. Algorithm performance with different types of vehicles

We first 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. 3 shows that the profit sum of the tours delivered by the proposed algorithm `approAlg` is about from 12.5% to 17.5% larger than those by algorithms `partitionAlg` and `forestAlg`. For example, the profit sums by algorithms `approAlg`, `partitionAlg` and `forestAlg` are 457.5, 362.4, 357.4, respectively, when the network size n is 200. Fig. 3 also shows that the profit sum by each of these three algorithms increases with the growth of the network size n. The rationale behind is that sensors are more densely located in a larger network, and the energy consumption of a charger on its traveling between sensors thus is less. Therefore, each charger has more energy for recharging sensors, thereby collecting more charging profit.

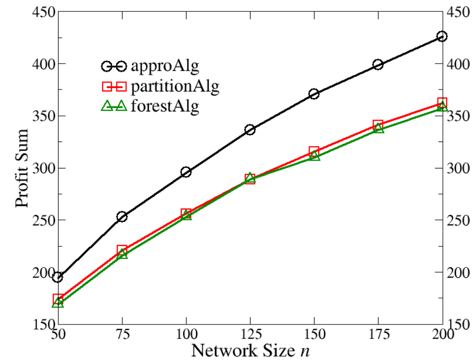


Fig. 3. Performance of different algorithms by varying the network size n from 50 to 200, when $K = 2$ and $B_{max} = 1,500$.

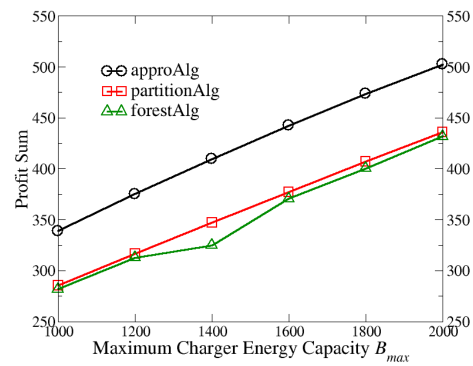


Fig. 4. Performance of different algorithms by varying the maximum energy capacity B_{max} from $1,000\ kJ$ to $2,000\ kJ$ while fixing B_{min} at $1,000\ kJ$, when $n = 200$ and $K = 2$.

We then study the algorithm performance, by increasing the maximum energy capacity B_{max} from $1,000\ kJ$ to $2,000\ kJ$ while fixing B_{min} at $1,000\ kJ$, where the energy capacity B_k of a charger k is randomly chosen from the interval $[B_{min}, B_{max}]$ with $1 \le k \le K$, $n = 200$ and $K = 2$. Fig. 4 plots the performance of algorithms `approAlg`, `partitionAlg` and `forestAlg`, from which it can be seen that the larger the maximum energy capacity B_{max} is, the higher the profit sum by each of these three algorithms is, since each charger can charge more energy-critical sensors when its energy capacity is larger. Fig. 4 also shows that the profit sum by algorithm `approAlg` is about from 15% to 18% larger than those by algorithms `partitionAlg` and `forestAlg`.

C. Algorithm performance with the same type of vehicles

We finally evaluate the performance of different algorithms with the same type of vehicles, by varying the energy capacity B of each charger from $2,000\ kJ$ to $5,000\ kJ$ when there are $n = 200$ sensors and $K = 2$ chargers in the network. Fig. 5 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 $3,500\ kJ$, while the profit sum by the former algorithm is upto 2% larger than that by the latter when

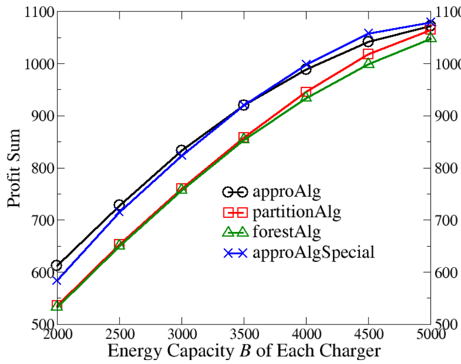


Fig. 5. Performance of different algorithms with the same type of chargers, by varying the energy capacity B of each charger from $2,000\ kJ$ to $5,000\ kJ$ when $n = 200$, $K = 2$, $\eta_k = 0.6\ kJ/m$ and $\rho_k = 0.95$.

B is larger than $3,500\ kJ$, 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 (see the last paragraph in the analysis of Lemma 3).

VI. Related Work

The orienteering problem and its variants have attracted a lot of attentions due to their wide applications [14], [27]. For the orienteering problem in metric graphs, Blum *et al.* [3] 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.* [6] recently further improved the ratio further 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.* [24] 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 *et al.* [7] 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 special team orienteering problem. Boussier *et al.* [4] proposed an exact algorithm for the problem. Their algorithm is only applicable to a small problem size, since the problem is NP-hard. Archetti *et al.* [1] devised a tabu search algorithm and a neighborhood search algorithm for the problem. Gavalas *et al.* [13] assumed that there is a profit of visiting a node only when the node is visited within its time window. They proposed meta-heuristic approaches. Yu *et al.* [32] considered the factor that the profits of visiting a node at different time points are different, and they devised a bee colony algorithm. Notice that the these meta-heuristic algorithms do not provide any performance guarantees on the delivered solutions, 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 team orienteering problem. Nemhauser *et al.* [23] 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 that is able to deliver a $(1-1/e)$-approximate solution, and the result is tight, where e is the base of the natural logarithm. They 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 [12]. Filmus *et al.* [11] improved the ratio to $1 - 1/e$ when $P = 1$. On the other hand, Buchbinder *et al.* [5] proposed a randomized approximation algorithm for the nonmonotonically 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.

VII. Conclusions

In this paper we studied the team orienteering problem with different types of vehicles. 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 no less than 0.32 when $\epsilon = 0.5$. For the special team orienteering problem with the same type of vehicles, we proposed an improved approximation algorithm. We finally evaluated the proposed algorithms with simulations, and experimental results showed that the proposed algorithms are very promising.

Acknowledgement

We appreciate the three anonymous referees for their constructive comments and valuable suggestions, which helped us improve the quality and presentation of the paper greatly.

The work by Wenzheng Xu was supported by the National Natural Science Foundation of China (NSFC) with grant number 61602330, and Sichuan Science and Technology Program (Grant No. 2018GZDZX0010). The work by Weifa Liang supported by the Australian Research Council Discovery Project (no. DP200101985). Also, the work by Jian Peng was supported by the National Key Research and Development Program of China (Grant No. 2017YFB0202403), Sichuan Science and Technology Programand (Grant No. 2017GZDZX0003), the Sichuan Key Research and Development Program (Grant No. 2019KJT0015-2018GZ0098), and Sichuan University–Zigong City cooperation fund (2018CDZG-15). Furthermore, the work by Tang Liu was supported by Scientific Research Fund of Sichuan Provincial Education Department (18ZA0404). Finally, the work by Sajal K. Das was partially supported by NSF grants CCF-1725755, CNS-1818942, and CNS-1545050.

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, 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.*, 2004, pp. 166–174.

[3] A. Blum, S. Chawla, D. 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, 2007.

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

[5] 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, 2015.

[6] C. Chekuri, N. Korula, and M. Pal, "Improved algorithms for orienteering and related problems," *ACM Trans. Algorithms (TALG)*, vol. 8, no. 3, article no. 23, 2012.

[7] K. Chen and S. Har-Peled, "The orienteering problem in the plane revisited," in *Proc. 22nd Annu. ACM Symp. Computat. Geometry*, 2006, pp. 247–254.

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

[9] (2017). *Drone Using in Hurricane IRMA*. [Online]. Available: http://thedronegirl.com/2017/09/16/7-ways-drones-helping-hurricane-irmaharvey-recovery-experts/

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

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

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

[13] 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, article no. 6, 2016.

[14] A. Gunawan, H. C. Lau, and P. Vansteenwegen, "Orienteering problem: A survey of recent variants, solution approaches and applications," *European J. Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[15] Y. Liang, W. Xu, W. Liang, J. Peng, X. Jia, Y. Zhou, and L. Duan, "Nonredundant information collection in rescue applications via an energy-constrained UAV," *IEEE Internet Things J. (IOT-J)*, vol. 6, no. 2, pp. 2945–2958, 2019.

[16] W. Liang, W. Xu, X. Ren, X. Jia, and X. Lin, "Maintaining large-scale rechargeable sensor networks perpetually via multiple mobile charging vehicles," *ACM Tran. Sensor Netw. (TOSN)*, vol. 12, no. 2, article no. 14, 2016.

[17] 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. (TON)*, vol. 25, no. 5, pp. 3161–3174, 2017.

[18] 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.

[19] 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. 39th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 624–633.

[20] 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. 38th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2018, pp. 331–320.

[21] 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, 2017.

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

[23] G. Nemhauser and L. A. Wolsey "An analysis of the approximations for maximizing submodular set functions-I," *Math. Programming*, vol. 14, pp. 265–294, 1978.

[24] A. Paul, D. Freund, A. Ferber, D. Shmoys, and D. Williamson, "Prize-collecting TSP with a budget constraint," in *Proc. 25th Annual European Symp. Algorithms (ESA)*, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017, pp. 62:1–62:14.

[25] 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 Int. Conf. Comput. Commun. (INFOCOM)*, 2020, to appear.

[26] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 14981511, Dec. 2016.

[27] P. Vansteenwegen, W. Souffriau, and D. V. Oudheusden, "The orienteering problem: a survey," *European J. Operational Research*, vol. 209. no. 1, pp. 1–10, 2011.

[28] C. Wang, J. Li, F. Ye, and Y. Yang, "A novel framework of multi-hop wireless charging for sensor networks using resonant repeaters," *IEEE Trans. Mobile Comput.*, vol. 16, no. 3, pp. 617–633, 2017.

[29] 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. (TMC)*, vol. 16, no. 4, pp. 990–1004, 2017.

[30] 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. (TMC)*, vol. 17, no. 11, pp. 2564–2577, 2018.

[31] 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. 39th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, 2019, pp. 881–890.

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

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

[34] Q. Zhang, W. Xu, W. Liang, J. Peng, T. Liu, and T. Wang, "An improved algorithm for dispatching the minimum number of electric charging vehicles for wireless sensor networks," *Wireless Netw. (WINE)*, vol. 25, no. 3, pp. 1371–1384, Apr. 2019.