On Minimum Time Multi-Robot Planning with Guarantees on the Total Collected Reward

Armin Sadeghi Ahmad Bilal Asghar Stephen L. Smith

Abstract— **In this paper, we study a multi-robot planning problem where a team of robots visit locations in an environment to collect a specified amount of reward in the minimum possible time. Each location has a weight associated with it that represents the value or amount of reward that can be collected by visiting that location. The problem is to design tours for the robots to collect at least D units of reward while minimizing the length of the longest robot tour. The single robot unweighted version of this problem is called the k-STROLL problem. We provide a 3-approximation algorithm for the multiple robot version of the k-STROLL problem. This leads to an algorithm for the weighted problem that collects at least $(1-\epsilon)D$ reward with tour lengths of at most 3 times the optimal tour length. The analysis of the approximation algorithm is then extended to provide bi-criterion approximations to two variations of the problem. We provide an application of the approximation algorithm for planning UAV tours with gimballed cameras for monitoring an urban environment.**

I. INTRODUCTION

In search and rescue applications, a fleet of robots are deployed to assess the risks and map the environment before deploying the rescuers. For instance, in the Fukushima nuclear disaster, a group of drones were deployed to scout the reactors and measure the intensity of radiation in the environment [1]. There is an inherent trade-off between the time required to map the scene and total information that is possible to be collected in that time. A key aspect in such applications is to recognize the set of locations for measurements that collectively provide enough information to map the environment and construct tours for the robots to visit the locations.

In this paper, we focus on the problem of deploying a set of robots to gather reward from locations in an environment. The robots' motion in the environment is captured as a roadmap (i.e., graph) with a reward $\phi(v)$ available on each vertex v of the graph. A group of robots, starting from a depot r, move on the common roadmap and collect reward from its vertices by visiting the locations of the vertices. The total reward collected is the cumulative reward available on the vertices visited by the robots. Depending on the application, the reward collection problem can be posed with three different objectives: 1) minimize the time to collect at least D reward with m robots, 2) minimize the number of robots to collect at least D reward with fixed battery life (i.e., operation time), and 3) maximize the total reward collected with m robots and fixed battery life.

This research is partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON, N2L 3G1 Canada (abasghar@uwaterloo.ca; a6sadegh@uwaterloo.ca; stephen.smith@uwaterloo.ca)

Related Work: The problem of maximal reward collection from an environment has been the subject of extensive research [2], [3], [4], [5] and appears in a variety of applications including 3D reconstruction of scenes [6] and ocean monitoring [7], [8]. Some of the informative path planning problems [9], [10] pose the problem as variations of an NP-hard problem known as the orienteering problem (OP) [11]. The orienteering problem consists of constructing a tour in a graph with weights on the vertices, representing the reward of each vertex, and a fixed budget on the tour length while maximizing the total reward of the vertices in the tour (i.e., the single robot version of problem 3 posed above). Among different solution approaches such as randomized algorithms [10], branch-and-cut algorithms [12], [13], the algorithm with strongest guarantee is a $(2+\epsilon)$-approximation algorithm by Chekuri *et. al.* [14]. For the version of the OP where the rewards are submodular instead of additive, a recursive greedy approximation algorithm is given in [15].

In [2], authors provide a constant factor approximation algorithm for the extension of the OP problem to multiple tours, also known as the *team orienteering* problem (TOP). In [16], authors study an extension to the TOP in which the weights on the vertices are not additive and the reward obtained from a vertex in a tour depends on the neighboring vertices of the vertex. A complete taxonomy of the recent solution approaches on different variations of the OP with benchmarking results is provided in [17]. The aforementioned studies address the problem of maximizing the reward collected while satisfying a bound on the total mission time. In contrast, our main focus in this paper is the problem where a bound is provided on the minimum total information required to map an environment and the objective is to minimize the mission time. The approximation algorithm for the TOP in [2] leverages the submodular property of the problem, whereas our problem does not possess submodularity, and requires a different solution approach.

A closely related problem is the k minimum spanning tree (k-MST) [18], [19] problem which is the problem of constructing a minimum cost tree that covers at least k vertices of the graph. A sequence of studies [18], [20], [21], [19] improve the guarantees on the approximation algorithms for the k-MST. The current best approximation ratio for the k-MST problem is 2 due to a primal-dual algorithm [19]. An extension to the k-MST problem is the k-stroll problem where the objective is to find the shortest path given a start and goal location visiting at least k vertices. The authors in [19] also provide a 2-approximation algorithm for the k-STROLL problem. In contrast to the existing results on the single k-STROLL, we consider the multi-robot and weighted version of the problem, where instead of visiting k vertices, the team of robots have to collect at least D reward from the graph. To the best of our knowledge, we provide the first approximation algorithm for this problem.

Contributions: The contributions of this paper are fourfold. First, we formulate the problem of minimizing the time to collect a fixed reward with multiple robots, namely the MIN-MAX GUARANTEED REWARD COLLECTION (MMGRC) problem (Section II). Second, we convert the problem to the unweighted version called the MIN-MAX k-STROLL problem and provide a 3-approximation algorithm for this problem, which leads to an algorithm for the weighted problem that collects at least $(1-\epsilon)D$ reward with tour lengths of at most 3 times the optimal tour length (Section III). Third, we extend the analysis of the approximation algorithm to give bi-criterion approximations for the problems of minimizing the number of robots required to collect a fixed reward and the TOP (Section IV). Finally, we provide an application for the MMGRC problem in an urban monitoring scenario (Section V). Table I shows the summary of our results.

II. PROBLEM FORMULATION

Consider a team of m sensor-equipped homogeneous robots that have to visit locations in an environment to collect reward. This reward, among other things, can represent the information available at different locations in the environment. The environment can be represented as a metric and undirected graph $G = (V, E, \ell)$ where the set of vertices represent the locations of interest in the environment. For each edge $e \in E$, the edge weight $\ell(e)$ represents the time it takes to traverse that edge and for each vertex $v \in V$, the vertex weight $\phi(v)$ represents the value or amount of the reward collected from that vertex.

All the robots start their tour from a common vertex r called the depot. Let T_i be the tour of robot i covering vertices $V_i \subseteq V$. With slight abuse of notation, we denote the length of the tour T_i by $\ell(T_i)$. The total reward collected from the environment is the sum of the rewards collected by individual robots on their tours, i.e., $\sum_{v \in \cup_{i=1}^m V_i} \phi(v)$. Note that once a robot collects reward from a location, subsequent visits to that location by any other robot will not contribute to the total reward collected. Also, to capture the time required to collect reward from a vertex, we add half of the reward collection time to all incident edges of that vertex which results in an equivalent metric graph with zero reward collection times. Hence, for the rest of the paper, we assume that no time is spent on reward collection.

In certain practical applications, the user has a requirement on the amount of information that must be gathered about the environment. For example, planning tours for imaging a scene for 3D reconstruction purposes may require a minimum amount of information necessary for an accurate 3D reconstruction. Therefore, we consider the problem of planning tours for the team of robots that collects at least D reward from the environment while minimizing the time required to collect that reward.

Problem II.1 (MIN-MAX GUARANTEED REWARD COLLECTION)**.** Given a metric graph $G = (V, E, \ell)$ with edge lengths $\ell(e), e \in E$ and vertex weights $\phi(v), v \in V$, along with m robots and a bound $D \in \mathbb{R}_+$, find a set of m tours $\{T_1, \ldots, T_m\}$ rooted at depot r such that $\sum_{v \in \cup_{i=1}^m V_i} \phi(v) \geq D$ and the objective function $\max_{i \in [m]} \ell(T_i)$ is minimized.

Observe that by setting $m = 1$ and $\phi(v) = 1, \forall v \in V$ in Problem II.1 we obtain the k-STROLL problem [19], which is known to be NP-hard by reduction from the travelling salesperson problem.

In the following section we present an approximation algorithm for Problem II.1. We will then extend the analysis of the approximation algorithm to provide bi-criteria approximation algorithms for two related problems.

III. APPROXIMATION ALGORITHM

We start by constructing an unweighted version of Problem II.1 with no weights on the vertices. Then we show that solving the unweighted problem will result in tours that collect at least $(1-\epsilon)D$ reward for some $\epsilon \in (0, 1)$.

A. Conversion to an unweighted problem

In this section, we convert the Problem II.1 of collecting at least D reward to the problem of visiting at least k vertices using a scaling method. The idea is similar to the scaling method used in the polynomial time approximation scheme for the knapsack problem in [22]. Given an instance of Problem II.1 with graph $G = (V, E, \ell)$, weight function $\phi : V \to \mathbb{R}_+$, m robots, depot $r \in V$ and some $\epsilon \in (0, 1)$, we construct a graph $G' = (V', E', \ell')$ as follows:

- Add r to V' and for each $v \in V \setminus \{r\}$ add a set F_v of $\lfloor \frac{\phi(v)|V|}{\epsilon D} \rfloor$ vertices to V'.
- For all $u, w \in F_v$ let $\ell'(u, w) = 0$.
- For all $v, z \in V$, $u \in F_v$ and $w \in F_z$ let $\ell'(u, w) = \ell(v, z)$.

Observe that the constructed graph G' is complete and the edge costs satisfy triangle inequality. Therefore, the problem of finding a set of m tours that collect at least D reward, i.e., Problem II.1 on graph G becomes the following problem on graph G':

Problem III.1 (MIN-MAX k-STROLL)**.** Given G', the number of robots m and a depot r, find a set of m tours, starting at r and covering at least $k = |V|(1/\epsilon - 1)$ vertices such that the length of the longest tour is minimized.

Note that the size of G' is polynomial in $|V|$ and $1/\epsilon$. We now give the following result relating Problems III.1 and II.1

Lemma III.2. *Given an $\epsilon \in (0, 1)$, there exists a set of m tours in $G = (V, E, \ell)$ that collects at least $(1-\epsilon)D$ reward such that the length of the longest tour is λ, if and only if there exists a set of m tours in $G' = (V', E', \ell')$ that visit at least $|V|(1/\epsilon - 1)$ vertices with maximum tour length of at most λ.*

Proof. Let $T_i^* = \langle r, v_1, \ldots, v_n \rangle$ be the tour of robot i in a solution of Problem II.1, then we construct a tour in G' with length at most $\ell(T_i^*)$ in G'. With the same sequence of vertices in T_i^*, for each $v_j \in T_i^*$, we add the vertices in F_v to the tour in G'. If the set F_v is empty, we remove it from the tour which results in a shorter tour by the metric property of G'. By constructing tours for each $i \in [m]$, the total number of vertices visited in G' is

$$\sum_{i\in[m]} \sum_{v\in T_i^*} \lfloor \frac{\phi(v)|V|}{\epsilon D} \rfloor \geq \sum_{i\in[m]} \sum_{v\in T_i^*} \frac{\phi(v)|V|}{\epsilon D} - 1$$
$$\geq |V|(1/\epsilon - 1),$$

where the second inequality is due to

$$\sum_{i\in[m]} \sum_{v\in T_i^*} \phi(v) \geq D.$$

Now for a given feasible set of m tours for Problem III.1 visiting at least $|V|(1/\epsilon - 1)$ in G', we construct a set of m tours of the same cost in G that collect at least $(1-\epsilon)D$ reward. Without loss of generality, we assume that if a tour in G' enters a set F_v then the tour visits all the vertices of F_v. Then given a tour $T_i = \langle r, v_1, \ldots, v_n \rangle$ in G', we construct a tour T_i' in G as follows: for each $v_j \in T_i$ we add vertex $v \in V$ to the tour T_i' such that $v_j \in F_v$. Therefore, the total collected reward in the constructed tours is

$$\sum_{i=1}^{m} \sum_{v\in T_i'} \phi_v \geq \frac{\epsilon D}{|V|} \sum_{i\in[m]} \sum_{v\in T_i'} \lfloor \frac{\phi(v)|V|}{\epsilon D} \rfloor \geq (1-\epsilon)D,$$

where the second inequality is due to $\sum_{i\in[m]} \sum_{v\in T_i'} \lfloor \frac{\phi(v)|V|}{\epsilon D} \rfloor \geq (1/\epsilon - 1)|V|$. □

Observe that due to Lemma III.2, a solution to Problem III.1 that covers at least k vertices in G' can be converted to a tour in G collecting at least $(1-\epsilon)D$ reward instead of D for Problem II.1.

B. *Algorithm for* MIN-MAX k-STROLL

The main challenge in this problem is to not only construct the tours minimizing the maximum length, but also to select the subset of vertices to visit. A naïve method would be constructing the minimum cost tree rooted at r that spans at least k vertices, then constructing a tour by doubling the edges of the tree and splitting the tour into m tours, one for each robot. However, this will not result in a constant factor approximation as illustrated in the example below. Figure 1 shows a graph with $2mq + 1$ vertices. The vertices are connected to their neighboring vertices with unit length edges, with the exception of r, which is connected by edges of length $1 + \delta$. Now consider the MIN-MAX k-STROLL problem on this graph with m robots visiting at least $k = mq$ vertices. Observe that the branch on the right-hand side of r is the minimum cost tree rooted at r and spanning mq vertices. With any tour construction and splitting method, the cost of the longest tour is at least $2mq$. However, the length of the longest tour in the optimal solution is $2(q + \delta)$ with each tour covering q vertices on a branch of the root node. For an arbitrarily small δ, this naive approach is an $\Omega(m)$ approximation algorithm.

The proposed approximation algorithm consists of a binary search on the length of the longest tour in the optimal solution, where at each step we construct a set of tours collectively covering at least k vertices. Algorithm 1 shows an iteration of the binary search and Algorithm 2 shows the complete approximation algorithm. Let λ be a guess on the length of the longest tour in the optimal solution. Then Algorithm 1 finds the induced subgraph of G where the vertices are within at most $\lambda/2$ distance from the depot. The function k-STROLL(G', r) in Line 3 of the algorithm utilizes an η-approximation algorithm for the k-STROLL problem to find a tour including r in G' that visits at least k vertices. The current best approximation ratio is $\eta = 2$ due to [19].

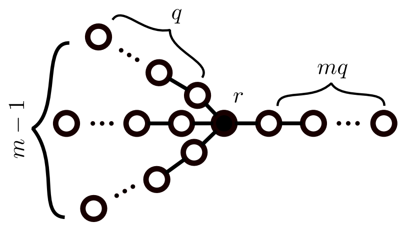

Fig. 1: An instance of the MIN-MAX k-STROLL problem with m robots visiting at least $k = mq$ vertices.

The SPLIT-TOUR subroutine splits the constructed tour T into tours of length at most $(1 + \eta)\lambda$. Let the tour T be $\langle r, v_1, v_2, \ldots, v_k \rangle$ visiting k vertices. For $i \in [m]$, let $s(i)$ and $t(i)$ be indices with $s(i) \leq t(i)$ such that the length of the tour $T_i = \langle r, v_{s(i)}, \ldots, v_{t(i)} \rangle$ is at most $(1 + \eta)\lambda$ and the length of the tour $\langle r, v_{s(i)}, \ldots, v_{t(i)}, v_{t(i)+1} \rangle$ is greater than $(1 + \eta)\lambda$. This tour splitting method is similar to the one used in [23] where a fixed number of tours is found to minimize the length of the largest tour. If the number of constructed tours are greater than the number of robots, then we increase λ and execute the same procedure with new λ. We can find the optimal λ using a binary search between 0 and $2k \max_{e\in E} \ell(e)$ as the length of the longest tour in the optimal solution is at most $2k \max_{e\in E} \ell(e)$.

Algorithm 1

1: **function** k-STROLLSPLIT$(G, k, \lambda, r, \alpha)$
2: $G' \leftarrow$ remove vertices v such that for the depot r, we have $\ell(r, v) > \lambda/2$
3: $T \leftarrow k-$STROLL(G', r)
4: $\mathcal{T} \leftarrow$ SPLIT-TOUR$(T, \alpha\lambda, r)$
5: **return** $\mathcal{T}$
6: **end function**

Algorithm 2

1: **function** MIN-MAX k-STROLL(G, m, k, λ, r)
2: Do a binary search in $[0, 2k \max_{e\in E} \ell(e)]$ to find the smallest value of λ such that k-STROLLSPLIT$(G, k, \lambda, r, 1 + \eta)$ returns m tours
3: **return** Return the tours obtained from k-STROLLSPLIT$(G, k, \lambda, r, 1 + \eta)$
4: **end function**

C. *Analysis*

Let $T_1^*, \ldots, T_m^*$ be the tours of m robots in the optimal solution and let λ^* be the length of the longest tour in the

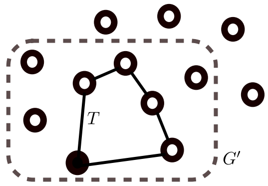

Fig. 2: Constructed tour on G' for $k=4$

optimal solution. The following result provides the worst-case performance of the proposed algorithm.

Proposition III.3 (Approximation factor)**.** *Suppose there exists an η-approximation algorithm for the k-*STROLL *problem, then the length of the longest tour obtained from the proposed algorithm is at most $(1+\eta)\lambda^*$.*

Prior to proof of the proposition, we provide the following observation on the vertices in the optimal solution.

Observation III.4. For any vertex v in $\cup_{i=1}^m T_i^*$, the distance of the vertex from the depot is $\ell(r,v) \leq \lambda^*/2$.

Proof. For any $i \in [m]$ and $v \in T_i^*$, by the triangle inequality, we have $2\ell(r,v) \leq \ell(T_i^*) \leq \lambda^*$. □

With this observation, we limit the set of candidate vertices by removing vertices farther that $\lambda/2$ from the depot. Observe that the SPLIT-TOUR subroutine divides tour T into tours of length at most $(1+\eta)\lambda$. Since connecting each vertex to the root adds at most $\lambda/2$ to the cost, each tour in $\mathcal{T}$ can cover at least $\eta\lambda$ of T.

Proof of Proposition III.3. We claim that for a λ, if $|\mathcal{T}| > m$, then $\lambda^* > \lambda$. Suppose that the opposite is correct, i.e., $\lambda^* \leq \lambda$. The length of the constructed tour T, using the k-stroll algorithm in [19], is $\ell(T) \leq \eta \sum_{i=1}^m \ell(T_i^*) \leq \eta m \lambda^*$. The SPLIT-TOUR method divides T into tours of length at most $(1+\eta)\lambda$. Each tour can cover at least $\eta\lambda$ of T, therefore,

$$|\mathcal{T}| \leq \left\lceil \frac{\ell(T)}{\eta\lambda} \right\rceil \leq \left\lceil \frac{\eta m \lambda^*}{\eta\lambda} \right\rceil \leq m.$$

This is a contradiction. □

Using the 2-approximation algorithm for k-STROLL from [19], we get a 3-approximation algorithm for MIN-MAX k-STROLL. Thus, to construct an approximation algorithm for the MMGRC, we first construct an instance of MIN-MAX k-STROLL as described in Section III.A. Then using Algorithm 2, we obtain a set of tours and construct a solution for MMGRC with the following guarantees.

Theorem III.5. *For any $\epsilon \in (0,1)$, the tours constructed for MMGRC using the 3-approximation for* MIN-MAX k-STROLL *problem collect at least $(1-\epsilon)D$ reward such that the length of the longest tour is at most 3 times the length of the longest tour in the optimal solution to MMGRC.*

The following remark gives the runtime of the proposed algorithm for MMGRC.

Remark III.6. (Runtime) Since the number of iterations in the binary search on λ is at most $O(\log(k \max_{e\in E} \ell(e)))$, and assuming that the runtime of the approximation algorithm for k-STROLL is $Q(n)$ on a graph of n vertices, the runtime of Algorithm 2 on $G = (V,E,\ell)$ is $O\big(Q(|V|/\epsilon)\log(k \max_{e\in E} \ell(e))\big)$, which is polynomial in the size of input to the problem for a fixed ϵ.

The main result of this section is summarized in Table I.

IV. VARIATIONS OF MMGRC

In this section, we extend the analysis of the approximation algorithm for Problem II.1 to provide some theoretical results for related problems. If the constraints on the number of robots and the minimum reward to be collected are not strict, which may be the case in real world applications, the trade-offs between different quantities of interest can be studied to come up with different solutions.

A. Minimize Number of Robots

If the number of robots is not fixed, we can consider a variation of MMGRC where the objective is minimizing the number of robots that collect the reward in a given time.

Problem IV.1 (Minimize Number of Robots)**.** Given a positive number D and the maximum tour length λ, find the minimum number of tours collecting at least D reward.

From our main result, given a $\beta \in \mathbb{Z}_+$, we can construct a set of βm^* tours of length at most $(1+2/\beta)\lambda$ collecting at least $(1-\epsilon)D$ reward, where m^* is the optimal number of robots in Problem IV.1. Given an instance of Problem IV.1, construct the unweighted instance of the problem as explained in Section III.A, then for a β, construct a set of tours using Algorithm 1 with $\alpha = 1+2/\beta$ and $k = |V|(1/\epsilon - 1)$. Finally, construct the set of tours $\{T_1,\ldots,T_p\}$ in graph G with the following guarantees.

Corollary IV.2. *The set $\{T_1,\ldots,T_p\}$ consists of $p \leq \beta m^*$ tours of length at most $(1+2/\beta)\lambda, \beta \in \mathbb{Z}_+$, and collect at least $(1-\epsilon)D$ reward.*

Proof. The proof follows directly from the fact that the length of the tour constructed by k-STROLL(G',r) is at most $2m^*\lambda$ [19]. By splitting the tour to tours of length at most $(1+2/\beta)\lambda$, each tour in the set $\mathcal{T}$ can cover at least length $(2/\beta)\lambda$ of T. Therefore,

$$|\mathcal{T}| \leq \lceil \frac{\ell(T)}{\frac{2}{\beta}\lambda} \rceil \leq \lceil \frac{2m^*\lambda}{\frac{2}{\beta}\lambda} \rceil \leq \beta m^*,$$

where the last inequality is due to integrality of β. □

B. Team Orienteering Problem

In the team orienteering problem, the objective is to maximize the reward collected in a given amount of time. The problem is formally defined below.

Problem IV.3 (Team Orienteering)**.** Given the number of robots m and the maximum tour length λ, find a set of m tours that maximizes the total reward collected.

An approximation algorithm for the problem is given in [2]. Here we provide two alternate solution approaches, that may be useful if the constraints are not hard and the increase in number of robots or maximum tour length may be tolerated for the benefit of additional reward collected. We also note that none of the approximation guarantees we provide and the approximation algorithm in [2] are strictly better than the other as they provide a trade-off between different constraints and the objective.

Proposition IV.4. *There exists an algorithm for Problem IV.3 that collects at least $(1-\epsilon)$ times the optimal reward using βm tours of length at most $(1+2/\beta)\lambda$, for $\beta \in \mathbb{Z}_+$.*

Proof. We use binary search to find the maximum reward that can be collected by βm tours of length at most $(1+2/\beta)\lambda$, where at each iteration of the binary search we use the result in Corollary IV.2. The reward collected is at least $(1-\epsilon)D^*$ where D^* is the reward collected by the optimal solution of Problem IV.3. □

We can also use an approximation algorithm for the single robot orienteering problem to get a bicriterion approximation algorithm for Problem IV.3. Let D^* be the reward obtained by the optimal solution to Problem IV.3 and let G' be the graph obtained by removing the vertices $v \in V$ such that $\ell(r,v) > \lambda/2$. Then we can extend the idea from Section III.B to get the following result.

Proposition IV.5. *A γ-approximation algorithm for the single robot orienteering problem implies an algorithm that constructs m tours rooted at r with a length of at most 2λ collecting at least D^*/γ reward.*

Proof. By Observation III.4, vertices in the optimal solution belong to G'. Moreover, given m optimal tours of length at most λ, we can construct one tour of length at most $m\lambda$ that covers all the vertices in the optimal tour. Hence, running the γ approximation algorithm for the single robot orienteering with tour length $m\lambda$ on the graph will give a tour that collects at least D^*/γ reward. Now, we can split the tour of length $\leq m\lambda$ into m tours rooted at r, each of length at most 2λ using the SPLIT-TOUR function. □

Using the $(2+\epsilon)$ approximation algorithm for single robot orienteering from [14], we get m tours of length at most 2λ that collect at least $D^*/(2+\epsilon)$ reward for $\epsilon > 0$. The results presented in this section are summarized in Table I.

V. Simulation results

In this section, first we evaluate the performance of Algorithm 2 both in time and solution quality on TOP instances [24], [25], then we provide an application of MMGRC to scene reconstruction. While MMGRC and TOP have different objectives, they have the same inputs, and thus we use the TOP library of instances for evaluation. Since the implementation of the approximation algorithm for single k-STROLL is not provided in [19], in this section we use an integer linear program (ILP) instead of the approximation algorithm to solve single robot k-STROLL in Line 3 of Algorithm 1. Observe that with an ILP solver for the single k-STROLL, we are able to solve the single k-STROLL optimally, making $\eta = 1$, and from Proposition III.3 this results in a 2-approximation for MMGRC.

TOP instances: Next, we evaluate the performance of proposed algorithm for MMGRC by comparing it with an ILP implementation of MMGRC on TOP instances [24], [25] with less than 100 vertices. The ILP used is a natural extension to the ILP for the OP in [26]. Table II shows the percentage difference between the longest tour length in the solution of our algorithm and the solution obtained from ILP for MMGRC with a 5 minute time limit. The time ratio represents the ratio of the ILP runtime to the runtime of the proposed algorithm. The number of vertices and the number of robots in each instance is given by n and m, respectively. In each instance, we set the first vertex to be the depot and the reward collected to be at least half of the total reward on the vertices of the instance, i.e., $D = 0.5\sum_{v\in V}\phi(v)$.

The length of the longest tour in the solution obtained from the proposed algorithm is within 27.2% of the solution obtained from ILP for MMGRC with 5 minutes of time limit. In the highlighted instances, the proposed algorithm provides a solution where the length of the longest tour is shorter than that of the ILP solution. As expected, in all the TOP instances, the proposed algorithm shows a significant improvement in the runtime compared to solving the full ILP.

Increasing the number of robots significantly increases the size of the ILP for MMGRC, therefore, the solution provided by the ILP within the 5 minute time limit might be sub-optimal. However, increasing the number of robots in the approximation algorithm changes the SPLIT-TOUR procedure, which is linear in the number of robots. Observe that with increasing the number of robots in the instance of the same size, the solution provided by the proposed algorithm improves with respect to the solution of ILP.

3D scene reconstruction: One application of the problem discussed in the paper is to plan paths for a team of UAVs to gather information from a certain scene in the environment for 3D reconstruction purposes. We demonstrate this by planning tours in an urban environment as shown in the Figure 3. The problem instance is created using a method similar to [6], where viewpoints are generated in a grid above the scene to be monitored. The scene represents buildings in the downtown of the city of Dallas, TX and is taken from [27]. For each of the viewpoints on the grid, 10 camera angles were randomly generated. The best angle was picked for each viewpoint using a score that was calculated assuming a square footprint of the camera, and then projecting rays onto the scene within that footprint and calculating each ray's score based on the distance and the angle of incidence. This scoring method encourages fronto-parallel viewing angles. Each ray's score is multiplied by the importance factor ζ of the building it is incident on. The color of the buildings in Figure 3 represents their importance factor. Since the information gathered from the environment is not additive, i.e., if two viewpoints are viewing the same part of the scene, the reward contribution due to that part should not be counted from both of those viewpoints. Hence, using a greedy approach, each plane in the scene contributes to the reward for only one viewpoint that views it best, i.e., has the highest score for that plane. The problem instance

TABLE I: Summary of results

Problem	Reward collected	Number of robots	Maximum tour length
MIN-MAX GUARANTEED REWARD COLLECTION (Problem II.1)			
Constraints and objective	$\geq D$	$\leq m$	λ^* (objective)
Theorem III.5	$\geq D(1-\epsilon)$	$\leq m$	$\leq 3\lambda^*$
MINIMIZE NUMBER OF ROBOTS (Problem IV.1)			
Constraints and objective	$\geq D$	m^* (objective)	$\leq \lambda$
Corollary IV.2	$\geq D(1-\epsilon)$	$\leq \beta m^*$	$\leq (1+\frac{2}{\beta})\lambda,\ \beta \in \mathbb{Z}_+$
TEAM ORIENTEERING PROBLEM (Problem IV.3)			
Constraints and objective	D^* (objective)	$\leq m$	$\leq \lambda$
Proposition IV.4	$\geq D^*(1-\epsilon)$	$\leq \beta m$	$\leq (1+\frac{2}{\beta})\lambda,\ \beta \in \mathbb{Z}_+$
Proposition IV.5	$\geq D^*/(2+\epsilon)$	$\leq m$	$\leq 2\lambda$

TABLE II: Performance of the proposed algorithm on TOP instances. n is the number of vertices and m is the number of robots. * shows the optimally-solved instances by the ILP for MMGRC.

Problem	m	% diff. in length	Time ratio
$n = 32$			
p.1.2.a	2	13.6	57
p.1.3.a	3	19.0	255
p.1.4.a	4	15.5	206
$n = 21$			
p.2.2.a*	2	5.6	347
p.2.3.a	3	14.5	2272
p.2.4.a	4	1.1	1948
$n = 33$			
p.3.2.a	2	23.4	739
p.3.3.a	3	11.0	652
p.3.4.a	4	2.6	911
$n = 66$			
p.5.2.a	2	**-5.9**	1
p.5.3.a	3	**-5.5**	17
p.5.4.a	4	**-46.8**	15
$n = 64$			
p.6.2.a	2	27.2	51
p.6.3.a	3	**-2.3**	53
p.6.4.a	4	**-5.6**	53

created using the method described above had 110 vertices with non-zero rewards, and another vertex was added on the ground to act as the depot for the UAVs. The MMGRC problem instance was to collect at least 80% of the reward from the environment using a team of 3 UAVs. The paths planned using our algorithm are shown in Figure 3. Notice that the blue and red UAVs do not visit some vertices with low rewards while on their way to higher reward vertices.

Figure 4 shows the reconstructed scene from the pictures taken by the UAVs. The average ray score incident on each face of the scene is represented by the gray intensity of that face. The darker shades of gray represent the faces that are reconstructed with higher detail. Note that the buildings with higher importance factors are reconstructed with more detail.

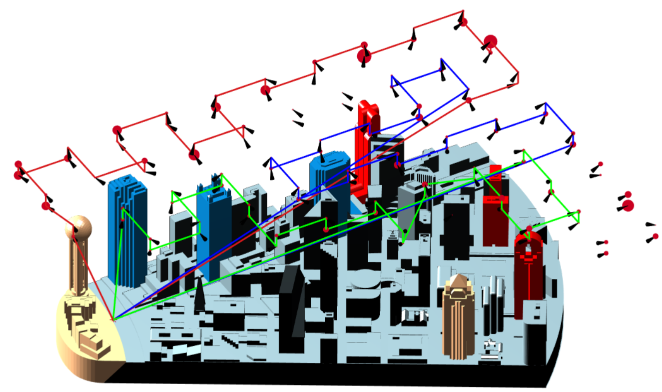

Fig. 3: Tours of three aerial robots taking pictures of the Dallas skyline for 3D reconstruction of the environment. The importance factors are $\zeta_{grey} = 1, \zeta_{blue} = 1.5, \zeta_{yellow} = 2, \zeta_{red} = 2.5$.

Fig. 4: Reconstructed scene using the tours in Figure 3.

VI. Conclusion and Future Work

This paper considered the problem of multi-robot reward collection. A constant factor approximation algorithm was proposed for the problem of minimizing the time to collect a guaranteed amount of reward. Then we proposed bi-criterion approximation algorithms to two related problems. The experiments on the TOP instances show significant improvement in the time to obtain near-optimal solutions. For future work, we hope to extend the results to capture multiple depots.

REFERENCES

[1] V. Beiser. (2018) The robot assault on Fukushima. [Online]. Available: https://www.wired.com/story/fukushima-robot-cleanup/

[2] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin, "Efficient planning of informative paths for multiple robots." in *International Joint Conference on Artificial Intelligence*, vol. 7, 2007, pp. 2204–2211.

[3] S. Alamdari, E. Fata, and S. L. Smith, "Persistent monitoring in discrete environments: Minimizing the maximum weighted latency between observations," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 138–154, 2014.

[4] M. Schwager, P. Dames, D. Rus, and V. Kumar, "A multi-robot control policy for information gathering in the presence of unknown hazards," in *Robotics research*. Springer, 2017, pp. 455–472.

[5] J. Binney, A. Krause, and G. S. Sukhatme, "Informative path planning for an autonomous underwater vehicle," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 4791–4796.

[6] M. Roberts, D. Dey, A. Truong, S. Sinha, S. Shah, A. Kapoor, P. Hanrahan, and N. Joshi, "Submodular trajectory optimization for aerial 3D scanning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5324–5333.

[7] R. N. Smith, M. Schwager, S. L. Smith, B. H. Jones, D. Rus, and G. S. Sukhatme, "Persistent ocean monitoring with underwater gliders: Adapting sampling resolution," *Journal of Field Robotics*, vol. 28, no. 5, pp. 714–741, 2011.

[8] K.-C. Ma, L. Liu, and G. S. Sukhatme, "An information-driven and disturbance-aware planning method for long-term ocean monitoring," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2102–2108.

[9] G. Best, J. Faigl, and R. Fitch, "Online planning for multi-robot active perception with self-organising maps," *Autonomous Robots*, vol. 42, no. 4, pp. 715–738, 2018.

[10] S. Arora and S. Scherer, "Randomized algorithm for informative path planning with budget constraints," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4997–5004.

[11] I.-M. Chao, B. L. Golden, and E. A. Wasil, "A fast and effective heuristic for the orienteering problem," *European journal of operational research*, vol. 88, no. 3, pp. 475–489, 1996.

[12] M. Fischetti, J. J. S. Gonzalez, and P. Toth, "Solving the orienteering problem through branch-and-cut," *INFORMS Journal on Computing*, vol. 10, no. 2, pp. 133–148, 1998.

[13] D. Feillet, P. Dejax, and M. Gendreau, "Traveling salesman problems with profits," *Transportation science*, vol. 39, no. 2, pp. 188–205, 2005.

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

[15] C. Chekuri and M. Pal, "A recursive greedy algorithm for walks in directed graphs," in *IEEE Symposium on Foundations of Computer Science*, 2005, pp. 245–253.

[16] J. Yu, M. Schwager, and D. Rus, "Correlated orienteering problem and its application to persistent monitoring tasks," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1106–1118, 2016.

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

[18] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala, "New approximation guarantees for minimum-weight k-trees and prize-collecting salesmen," *SIAM Journal on computing*, vol. 28, no. 1, pp. 254–262, 1998.

[19] N. Garg, "Saving an epsilon: a 2-approximation for the k-MST problem in graphs," in *Annual ACM symposium on Theory of computing*, 2005, pp. 396–402.

[20] S. Arya and H. Ramesh, "A 2.5-factor approximation algorithm for the k-MST problem," *Information Processing Letters*, vol. 65, no. 3, pp. 117–118, 1998.

[21] S. Arora and G. Karakostas, "A 2+ ε approximation algorithm for the k-MST problem," in *Annual ACM-SIAM symposium on Discrete algorithms*, 2000, pp. 754–759.

[22] S. Dasgupta, C. H. Papadimitriou, and U. V. Vazirani, *Algorithms*. McGraw-Hill Higher Education, 2008.

[23] G. N. Frederickson, M. S. Hecht, and C. E. Kim, "Approximation algorithms for some routing problems," in *IEEE Symposium on foundations of computer science*, 1976, pp. 216–227.

[24] I. Chao, "Algorithms and solutions to multi-level vehicle routing problems," Ph.D. dissertation, University of Maryland at College Park, 1993.

[25] The Orienteering problem: Test instances. [Online]. Available: https://www.mech.kuleuven.be/en/cib/op#section-3

[26] A. N. Letchford, S. D. Nasiri, and D. O. Theis, "Compact formulations of the steiner traveling salesman problem and related problems," *European Journal of Operational Research*, vol. 228, no. 1, pp. 83–92, 2013.

[27] B. Warman. (2016) Dallas Skyline. [Online]. Available: https://www.thingiverse.com/thing:1516990