Research Paper Summary: Team Orienteering Planning Coverage with Uncertain Reward

Brandyn Tucknott

Last Updated: 25 October 2025

1 Introduction to TOCPUR

Mobile agent fleets are used daily, and we often wish to collect as much reward as possible while staying within some allotted time or budget limit. This can be formulated as a Team Orienteering Problem (TOP), however TOP assumes the reward at each location is a known constant. This assumption is not true in general, and such as in cases when the reward can be accumulated over time. For example, consider a fleet of garbage trucks tasked with picking up garbage throughout a city. The amount of garbage at each location is not fixed, but grows over time. In addition to this, the amount of garbage at each location is unknown to the agent until they arrive. To compensate for this, there is an expected garbage growth rate associated with each location, and after every visit, this rate is updated depending on the results on arrival. Finally, TOP typically has the limitation that each location can only be visited once, limiting its application.

To avoid these issues, a new class of problem is introduced: Team Orienteering Coverage Planning with Uncertain Reward (TOCPUR). In TOCPUR, a team of mobile agents is tasked to patrol a set of predetermined locations over multiple iterations. The overall goal is to reduce time and place-dependent costs, and in particular, we assume cost grows between iterations and stays constant within each iteration. Each vertex can be visited multiple times in an iteration, but cost is only reduced once. Optionally, a subset of mandatory locations can be specified. To solve TOCPUR, estimate unkown and growing costs over an area and solve a variant of TOP: Team Orienteering Coverage Problem (TOCP). TOP and TOCP plans refer the solutions for their corresponding problems.

2 Background

This section introduces notation and formally defines the TOCPUR problem.

2.1 Notation

The area to be patrolled is represented with a symmetric directed graph $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ is the set of N verticies and E is the edge set. e_{ij} denotes an edge going from vertex v_i to v_j, which has a length of l_{ij}. We assume the agent fleet consists of M agents. We use $[x]$ to denote $\{1, \ldots, x\}$, and $[a, b]$ to denote $\{a, a+1, \ldots, b\}$.

2.2 Problem Formulation

Here we define the TOCPUR problem. Consider H iterations of route planning for a fleet of size M over a symmetric diracted graph $G = (V, E)$. Each vertex v_i accumulates a cost $\kappa_{i,t}$ right before the t-th iteration, where $\mathbb{E}(\kappa_{i,t}) = \mu_i^*$ is unkown (The cost at each vertex grows approximately linearly) [In the garbage truck example, think of the iteration as a day, and $\kappa_{i,t}$ as the amount of garbage that appears at a location overnight]. Denote $T_{i,t}$ as the most recent iteration prior to t when the fleet visited v_i. Then v_i at iteration

t accumulates a total cost of

$$c_{i,t} = \sum_{k=T_{i,t}+1}^{t} \kappa_{i,k}.$$

The entire cost over G at iteration t is specified as

$$c(G,t) = \sum_{i\in[N]} c_{i,t}.$$

The total cost at the end of an iteration is the sum of accumulated costs of verticies which were not visited during that iteration (cost of visited verticies is reset to 0). The goal then, is to plan M routes $\{\tau_{m,t}\}_{m=1}^{M}$, each no longer than a maximum length $l_{\max}$ for all vehicles at each iteration $t \in [H]$, such that the total cost on G over the horizon is minimized. Specifically, each route $\tau_{m,t} = (v_1, \ldots, v_1)$ is a sequence of verticies starting and ending at v_1. We can optionally include a set of "must visit" verticies I such that the fleet has to visit all verticies in I during each iteration. The whole problem can be described as the following optimization:

$$\begin{aligned}
\underset{\{\{\tau_{m,t}{}_{m=1}^{M}\}\}_{t=1}^{H}}{\text{minimize}} \quad & \mathcal{L} = \mathbb{E}\left(\sum_{t=1}^{H} c(G,t)\right) \text{ where} \\
& c(G,t) = \sum_{i=1}^{N}\sum_{k=T_{i,t}+1}^{t} \kappa_{i,k}. \\
\text{subject to } & \forall_{m,t} \sum_{e_{i,j}\in\tau_{m,t}} l_{ij} \le l_{\max}, \\
& \forall_{i,t} T_{i,t+1} = \begin{cases} t, & \text{if } v_i \in \cup_m \tau_{m,t} T_{i,t}, \\ \text{otherwise} & \end{cases} \\
& \forall_i T_{i,1} = 0 \\
& \text{optionally } \forall_{v_i\in I} v_i \in \cup_m \tau_{m,t}.
\end{aligned}$$

Here, $\cup_m \tau_{m,t} = \cup_m \{v : v \in \tau_{m,t}\}$ and we abuse the notation to denote $e_{ij} \in \tau$ if v_i or v_j appear consecutively in τ. This optimization problem (Opt. 1) is NP hard even if we know $\{\mu_i^*\}$ since it reduces to the Travelling Salesman Problem. There are two important points:

1. Since $\kappa_{i,t}$ is drawn randomly from a distribution with an uknown mean μ_i^*, it is generally impossible to have an open-loop plan to solve Opt. 1
2. Unlike in TOP, we do not assume each v_i can only appear once in any $\tau_{m,t}$. We can however, safely assume each e_{ij} appears at most once in any $\tau_{m,t}$

3 Method

The proposed method to solving the TOCPUR problem is by optimizing the per-iteration objective while simultaneously updating the reward estimation.

Reward Estimation and Per-Iteration Planning Solving Opt. 1 for large H is infeasible; in fact, even when $H = 1$ and $\{\mu_i^*\}_{i=1}^{N}$ are known, the problem is still NP hard. Since $\{\mu_i^*\}_{i=1}^{N}$ are unknown, an optimal solution must consist of closed loop plans that take past observations (κ) into consideration. The solution then, is to optimize the expected cost of $\mathbb{E}\left(c(G,t)\right)$ iteratively while updating $\{\mu_i^*\}_{i=1}^{N}$ simultaneously. In particular, we keep track of $T_{i,t}$ for each node v_i and the total observed cumulative cost at v_i, i.e. $C_{i,t} = \sum_{k=1}^{T_{i,t}} \kappa_{i,t}$. Then the maximum liklihood estimation of μ_i^* at time t is

$$\hat{\mu_{i,t}} = \begin{cases} \frac{C_{i,t}}{T_{i,t}} = \frac{-sum_{k=1}^{T_{i,t}}\kappa_{i,k}}{T_{i,t}}, & T_{i,t} > 0 \\ \mu_{\text{default}}, & {}_{i,t} = 0, \end{cases}$$

where μ_{default} is a default or pre-specified value based on prior knownledge when there were no visits to the node. At each iteration t, $\{\hat{\mu_{i,t}}\}_{i=1}^{N}$ is used to estimate $\mathbb{E}\left(c(G,t)\right)$, i.e. $\hat{c} = \sum_{i=1}^{N} \hat{\mu_{i,t}} \left(t - T_{i,t}\right) \approx \sum_{i=1}^{N} \mu_i^* \left(t - T_{i,t}\right)$.

3.1 Greedy Per-Step Planning