Extended Proofs for "Relay Pursuit for Multirobot Target Tracking on Tile Graphs"

Shashwata Mandal[1] and Sourabh Bhattacharya[1,2]

The document provides a detailed proof of properties of tile graphs, tile cover and minimal tile cover used in the submitted version of ICRA 2023 paper titled "Relay Pursuit for Multirobot Target Tracking on Tile Graphs".

A. Minimal Tile Cover Computation

Given an environment, a Minimal Tile Cover (MTC) is a tile cover of minimum cardinality.

MTCP. *An Apriori Structure is defined as a set of sets* $W = \{T_1, ..., T_u\}$ *where* $T_i \subseteq Q$ *where* $Q = \{c_1, \ldots, c_n\}$*. Given a Apriori Structure* W *and corners* Q*, what is the minimum size of a disjoint cover* F *such that* $x \in F \wedge x \subseteq T_p \wedge T_p \in W$ *and* $Q = \bigcup_{y \in F} y$*?*

Theorem 1. *MTCP is NP-Hard.*

Set Cover. *Given a universe of elements* $U = \{e_1, \ldots, e_m\}$ *and a set of sets* $\mathcal{S} = \{S_1, \ldots, S_t\}$ *where* $S_i \subseteq U$*, what is the minimum size of a set* $\mathcal{C}$ *such that* $\mathcal{C} \subseteq \mathcal{S}$ *and* $U = \bigcup_{j \in \mathcal{C}} j$*?*

Proof. Minimal Tile Cover Problem (MTCP) can be shown to be NP-Hard by reducing an instance of Set Cover (a known NP-Hard problem [1]) to it. An instance of Set Cover can be represented as $\langle U, \mathcal{S} \rangle$ and an instance of MTCP can be represented as $\langle Q, W \rangle$. To map an instance of Set Cover to an instance of MTCP, $U \mapsto Q$ since both describe the universe of elements. The Apriori Structure W is also a set of sets similar to $\mathcal{S}$. So $\mathcal{S} \mapsto W$. This can be done in polynomial time. For the reverse mapping, when we get a MTCP G (set of disjoint sets), the sets in G can be traced back to its superset in W. These sets are a part of $\mathcal{S}$ and this operation doesn't change the size of the cover and is also in polynomial time. Since, G covers all elements in Q and each of these elements can be mapped to a set in $\mathcal{S}$, we can guarantee that the resulting set cover is the minimum set cover. Hence, $\langle U, \mathcal{S} \rangle \mapsto \langle Q, W \rangle$. Since Set Cover is a known NP-Hard problem, MTCP has to be NP-Hard. ∎

Algorithm 3 uses a greedy strategy to generate a tile cover on an Apriori structure. Since higher levels of the Apriori structure combine more corners compared to lower levels, the algorithm greedily iterates from the top to the bottom of the Apriori structure. In Algorithm 3, U denotes the set of tiles that form the tile cover. At each level of the apriori structure, we add a set of disjoint tiles to U that maximizes its cardinality. This is done as follows. F denotes the set of tiles at a given level that does not have any corners already present in U (Line 5). Next, we construct a graph $G = (V, E)$ such that V is a set of tiles in F. An edge exists between two vertices if the corresponding tiles have a non-empty intersection (Line 9). Initially, all vertices are marked as *valid*. Next, vertices of G are scanned in the increasing order of their degree. At any stage, a vertex marked valid is added to U, and its neighbours are marked *invalid* before removing the vertex from G. This process continues till either G is empty or all vertices of G are marked invalid.

Algorithm 3 Algorithm for Tile Cover

Input: k Height of structure, L Set of items set till k
Output: U Tile Cover

1: **function** MINIMIZE TILES(L, k)
2: $U \leftarrow \phi$
3: **while** $k \geq 0$ **do**
4: **for** $i \in L_k$ **do**
5: **if** $\forall x \in i, \nexists y \in U \wedge x \notin y$ **then**
6: Add i to F
7: **end if**
8: **end for**
9: Create a graph $G = (V, E)$ s.t. $V = \{v | \forall v \in F\}$ and $E = \{(u, v) | u, v \in F \wedge u \cap v \neq \phi\}$.
10: $j \leftarrow 0$
11: **while** *valid* nodes remain in G **do**
12: Add nodes with degree j to U and set linked nodes to *invalid*
13: $j \leftarrow j + 1$
14: **end while**
15: $k \leftarrow k - 1$
16: **end while**
17: **return** U
18: **end function**

B. Tight bound on the number of guards and their speed

Theorem 2. *The minimum speed for required to track an evader for* $m-1$ *guards is* v_e

[1]Department of Computer Science, Iowa State University, Ames, IA 50010, USA, [2]Department of Mechanical Engineering, Iowa State University, Ames, IA 50010, USA smandal@iastate.edu, sbhattac@iastate.edu

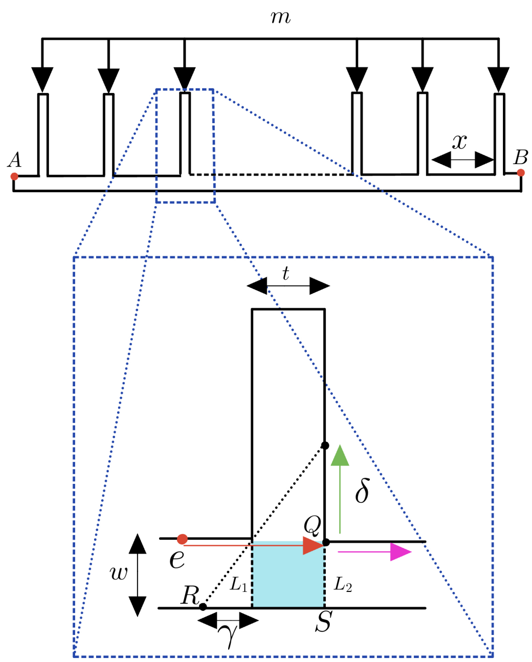

Fig. 1: A "comb" environment

Proof. Let $O = \{o_1, ..., o_k\}$ be a set of guards such that $k < m$, where m is the cardinality of the minimum tile cover. Let v_p and v_e be the speed of the observers and evader respectively such that $v_e = rv_p$ where $r > 1$. The polygon in Figure 1 has m teeth (vertical rectangular structures) each separated by a distance x. Let t be the width of each tooth, w be the width of the hallway (horizontal rectangular structure) and the extensions of the left and right walls of the tooth be L_1 and L_2 respectively. Each tooth induces a star tile. If a static guard is placed on a star tile (blue region) of each tooth, the evader cannot escape i.e. when $k = m$. However, when $k < m$, some teeth will remain uncovered hence k static guards cannot prevent the escape. Now we show that even dynamic guards cannot prevent the escape when the evader follows the following strategy. The evader travels in a straight line from A to B and tries to escape via L_2 at Q of every tooth (green arrow) if an escape is possible. If the evader e travels a distance δ along L_2, we get $\gamma = \frac{tw}{\delta}$ via symmetry. γ represents the maximum distance from the line from L_1 at which the guard can still see the evader. Since $\gamma \propto tw$ if we reduce tw we can reduce the field of visibility of the evader from the hallway. If the evader tries to escape via L_2, no guard can see the evader right of the line L_2 and has to be within γ distance left of L_1 to see the evader. If $tw \lll \delta$, R is very close to L_1. At each tooth, the evader decides at point Q whether it can escape via L_2, so a guard has to be between R and S to prevent the escape. If the guard is at L_2 or behind it, the evader can leave the guard and move towards the next tooth (pink arrow) since $v_e > v_p$. So the same guard cannot be responsible for tracking the evader in the next tooth. If a guard from a previous tooth wanted to reach R then it would need to travel from L_2 of the previous tooth to R of the current tooth when evader travels form Q of the previous tooth to the Q of the current tooth. The minimum value of x to prevent this can be solved by

$$
\begin{aligned}
r &= \frac{x+t}{x-\gamma} \\
\implies x &= \frac{t+r\gamma}{r-1}
\end{aligned} \tag{1}
$$

Then if $x > \frac{t+r\gamma}{r-1}$, the previous guard cannot keep up with the evader. So we can show that at each tooth, any guard behind it cannot interfere with its escape around the current tooth if $tw \lll \delta$ and $x > \frac{t+r\gamma}{r-1}$. After each tooth, a guard is left behind by the evader so, after j teeth, j guards are behind the evader i.e. $k - j$ guards are remaining to guard the rest of the teeth. Then after k teeth, there are no more guards left since $k < m$ hence the evader can escape. Therefore, we can say the evader cannot be tracked for any guard speed less than v_e no matter what the policy. ■

References

[1] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, 1972, pp. 85–103. [Online]. Available: https://doi.org/10.1007/978-1-4684-2001-2_9