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 expanded*

Minimal Tile Cover is the minimum tile cover given an Apriori structure till L_h. The height of the structure is the maximum level of iterations the structure can run for till no further itemsets can be generated represented by h. The Minimal Tile Cover will be represented by MTC. It turns out MTC is NP-Hard. The following algorithm generates a tile cover using a greedy strategy.

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**

Theorem 1. *Minimal Tile Cover(MTC) is NP-Hard*

MTC. *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 $Q = \bigcup_{y \in F} y$?*

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 can shown to be NP-Hard by reducing an instance of Set Cover (a known NP-Hard problem) to it. An instance of Set Cover can be represented as $\langle U,\mathcal{S}\rangle$ and an instance of MTC can be represented as $\langle Q,W\rangle$. To map an instance of Set Cover to an instance of MTC, $U \mapsto Q$ 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 MTC 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, then MTC has to be NP-Hard. ■

The MTC problem runs on a structure that is exponential in size of the corners to begin with. However it turns out finding the optimal given an Apriori Structure is NP-Hard. However, when we are using Apriori to solve the problem of intersecting corners we are restricted to small input sizes since the base problem still remains exponential. So we need to generate a tile cover which attempts to minimize the size of the tile cover. Algorithm 3 generates 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: We do this by first creating F(set of tiles excluding corners already in C_k) (Line 5) to prevent corner repetition. Then we construct a graph $G = (V,E)$ such that V is a set of tiles in F and E records intersections between those tiles (Line 9). All nodes are marked as *valid*. Nodes of G are extracted from degree 0 till $k-1$ until all *valid* nodes are gone. After each node v is extracted and added to U, all nodes such

[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

that $u \in V \wedge (u,v) \in E$ are marked *invalid*. Once level k is computed, the algorithm moves to level $k-1$.

Theorem 2. *Tile cover forms a connected graph.*

Proof. Let D be a tile cover of a polygon P. Then $D = \{T_1, \ldots, T_m\}$, such that $\bigcup_{s \in D} s = \{c_1, \ldots, c_n\}$. Let G be a connected graph of tiles in D. Two tiles T_1 and T_2 are said to have mutual visibility if $\exists p \in T_1, q \in T_2$ s.t. pq is a line completely contained in the polygon. We say that the line of sight between T_1 and T_2 exists if they have mutual visibility. Let $G = (V, E)$ be a graph where each tile in D forms a vertex in G s.t. $|V| = |D| = m$. Two vertices $u, v \in V$ form an edge (u,v) in G if they have mutual visibility. A graph is said to be connected if it has one connected component. For a vertex u to be added to H, u needs to have an edge with $v \in H$ where H is a connected component. Let us start with a single vertex $v_1 \in V$ in a connected component H and try to add a vertex to H at each step. Now, let us consider two vertices $u, v \in V$ such that $v \in H$ and $u \notin H$. If v has mutual visibility with u, u is added to C. If v does not have mutual visibility with u, then the line of sight between u and v has to be obstructed by some corners in P. Let L be the set of corners that obstructs the mutual visibility between u and v. Then, c is the one closest corner to the tile T_v corresponding to v. Let T_w be the tile associated with c where $w \in V$. Since c is the closest corner breaking line of sight for T_v, then T_w and T_v have mutual visibility. So w is added to H. Since at add each step a vertex is added to the connected component, the entire graph gets connected in $m-1$ steps. ■

Theorem 3. *The worst-case minimum speed independent of the geometry of the polygon for a group of k guards is v_e if $k < |MTC|$.*

Proof. Let $O = \{o_1, ..., o_k\}$ be the number of observers such that $k < m$ where m is the cardinality of the minimum tile cover. Let us assume that there exists a speed ratio $r > 1$ for k observers to track an evader e. Let v_p and v_e be the speed of the observers and evader respectively such that $v_e = rv_p$. The polygon in Figure 1 has m fringes each separated by arbitrary distances. Each fringe induces a star tile. If $k = m$, the whole polygon would be covered. However $k < m$, so there aren't enough tiles to cover all the fringes. Let the width of these fringes be some very really small finite value. Let us consider the following trajectory - the evader starts at the left-most fringe and travels to the right-most fringe trying to evade at each fringe. Let the fringes be labeled 1 to m. We can see that a guard needs to be placed at the tile associated with the fringe to prevent the escape of the guard. If a guard is covering a fringe, since the guard is slower than the evader, the same guard cannot keep up with the evader till it tries to cover the neighboring fringe. If the guard is currently at the j numbered fringe, we can guarantee that j guards have already been used since the evader would have escaped. So the number of remaining guards is $k-t$ which is less than the remaining number of fringes. So at the k-th fringe, the evader is guaranteed to escape. Hence, we can say the evader cannot be tracked for any guard speed less than v_e no matter what the policy. ■

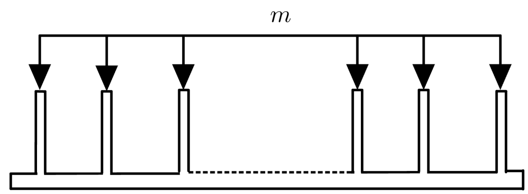

Fig. 1: A "comb" environment