COMPUTING STRATEGIES FOR MULTIPLE COPS ON INFINITE GRAPHS

ALEXA MCLEOD AND REED SOLOMON

1. Introduction

The game of cops and robbers was introduced independently in Quilliot [14] and in Nowakowski and Winkler [13]. The original game is played on a graph G by two players: the cop and the robber. The game begins with the cop, followed by the robber, choosing a vertex on which to start. In subsequent rounds, the players take turns (with the cop going first) moving from their current vertex to one of the neighboring vertices. The game ends, and the cop wins, if she shares a vertex with the robber at any point in the play. The robber wins if he manages to avoid ever sharing a vertex with the cop.

There are two standard assumptions made on the graphs used in this game. First, the robber can always win when the game is played on a disconnected graph by choosing a starting node in a different component from the cop's starting position. To avoid this triviality, we assume our graphs are connected. Second, to allow the players to choose to remain on their current vertex in any given round, we assume the edge relation is reflexive. We make the additional assumption here that our graphs are at most countable because our eventual concern is with computable graphs and strategies.

Cops and robbers is an open game, and therefore, for any fixed graph G, one of the players has a winning strategy when the game is played on G. G is called *cop-win* when the cop has a winning strategy and *robber-win* otherwise (when the robber has a winning strategy). Nowakowski and Winkler [13] gave two structural characterizations of cop-win graphs. The first characterization applies to finite graphs and shows there is a polynomial time algorithm to determine if a finite graph is cop-win. The second characterization applies to all graphs and was used by Stahl [16] to show the index set of computable cop-win graphs is Π^1_1-complete.

From the perspective of computability theory, we would like to understand the complexity of strategies required to win this game on computable graphs. It is not difficult to use trees to code information into robber-win strategies. For each computable ordinal α, Stahl [16] gave an example of a computable graph that is robber-win and such that every robber-win strategy computes $0^{(\alpha)}$. It is more challenging to deal with cop-win strategies. Nonetheless, Stahl [16] constructed a computable graph that is classically cop-win but not by a computable cop strategy.

In this paper, we extend these results to a variation of cops and robbers in which there are multiple cops. The version with n cops is played in rounds as before. In the initial round, the player controlling the cops chooses a starting vertex for each cop, after which the player controlling the robber chooses a starting vertex for the robber. In the subsequent rounds, each cop moves to a neighboring vertex, followed by the robber moving to a neighboring vertex. The cops win if any cop shares a vertex with the robber at any point in the play, and otherwise, the robber wins.

The game with n cops is still an open game, so for any n and G, there is either a winning strategy for the n cops (and G is *n-cop-win*) or there is a winning strategy for the robber (and G is *robber-win* or *not n-cop-win* to emphasize the number of cops in the game). Note that if $n = |G|$, then G is n-cop-win because the player for the cops can place one cop on each vertex in the initial round.

The minimum number of cops required to win on a fixed graph G is called the *cop-number* of G, denoted $c(G)$. This notion is well studied for finite graphs and there are many results in the literature giving upper and lower bounds for various classes of graphs. For example, $c(G) \leq 3$ for finite planar graph G (by Aigner and Fromme [1]), while $c(G) \leq 2$ for finite outerplanar graphs (by Clarke [7]).

In Section 2, we consider the complexity of strategies in the game with n cops on computable graphs. We show that Stahl's examples for coding information into robber strategies continue to work in this context. More interestingly, for each $n \geq 2$, we construct a computable graph G such that G is cop-win classically, G can be won by n cops following a computable strategy, but G cannot be won by $n-1$ cops following a computable strategy. Thus we can achieve any desired "computable cop-number" with a graph which is classically cop-win.

In Section 3, we show that for each $n \geq 2$, the index set of computable graphs that are n-cop-win is Π^1_1-complete. The hardness half of this result comes directly from Stahl [16], but to show that the index set is Π^1_1, we use a structural characterization of n-cop-win graphs by Clarke and MacGillivray [8].

Our notation for computability theory is standard and follows Soare [15] and Ash and Knight [2]. Bonato and Nowakowski [5] and Bonato [4] are excellent introductions to cops and robbers and to vertex pursuit games (and other games on graphs) more generally.

2. Computable strategies

We consider the game of cops and robbers with n cops on a reflexive connected graph G with vertices $\{v_n \mid n \in \omega\}$. For determining whether G is n-cop-win or robber-win, there is no loss of generality in requiring all the cops to start the game at v_0. That is, suppose the cops are initially placed at v_0, but a cop-winning strategy wants the cops to start elsewhere. Since G is connected, each cop can move along a path from v_0 to their desired starting vertex in the opening rounds of the game. Once all of the cops have reached their locations, they can follow the cop-winning strategy.

To formalize the definition of a strategy, we denote the position of the robber in round i by $r_i \in \omega$ (indicating that the robber is on v_{r_i}) and the positions of the n cops by a string $\gamma_i \in \omega^n$ (indicating that the k-th cop is on $v_{\gamma_i(k)}$). Since we assume the cops start at v_0, we have $\gamma_0 = 0^n$.

The history of the game at any point can be represented by a tuple of the form $\langle \gamma_0, r_0, \gamma_1, r_1, \ldots, \gamma_j \rangle$ or $\langle \gamma_0, r_0, \gamma_1, r_1, \ldots, \gamma_j, r_j \rangle$, depending on whether the cops or the robber moved last. The sequence represents a legal play as long as $\gamma_0 = 0^n$ and there are edges between v_{r_i} and $v_{r_{i+1}}$ and between $\gamma_i(k)$ and $\gamma_{i+1}(k)$ for each $k < n$. The sequence represents a cop win if there is an i and $k < n$ such that $r_i = \gamma_i(k)$ or $r_i = \gamma_{i+1}(k)$.

Coding sequences by numbers, the states of the game become elements of ω. An *n-cop strategy* on a graph G is a function $f : \omega \to \omega$ that takes (a code for) a state of the game as input and outputs an allowable cop move. More formally, if m represents a legal state of the game $\langle \gamma_0, r_0, \ldots, \gamma_j, r_j \rangle$, then $f(m)$ is (the code for) an n-tuple τ such that $\langle \gamma_0, r_0, \ldots, \gamma_j, r_j, \tau \rangle$ is a legal move.

We say that a (finite or infinite) play of the game $\gamma_0, r_0, \gamma_1, r_1, \ldots$ *follows the n-cop strategy f* if $\gamma_0 = 0^n$ and $\gamma_{i+1} = f(\langle \gamma_0, r_0 \ldots, \gamma_i, r_i \rangle)$ for each i. An n-cop strategy f is *winning* if the cops win every legal play that follows f.

Analogously, a *robber strategy* on G is a function $f : \omega \to \omega$ that outputs an allowable robber move when it is the robber's turn. The play $\gamma_0, r_0, \gamma_1, r_1, \ldots$ *follows the robber strategy f* if $r_i = f(\langle \gamma_0, r_0 \ldots, \gamma_i \rangle)$ for each i. A robber strategy f is *winning* if the robber wins every legal play that follows f.

Note that the set of strings representing legal plays on a computable graph G is computable. A *computable n-cop strategy for G* is a total computable function f that is an n-cop strategy for G.

We use known results to code information into robber strategies. We can view a tree $T \subseteq \omega^{<\omega}$ as a graph by letting the elements of T be the nodes in the graph and putting an edge between each node and its immediate successors on T (e.g. between σ and $\sigma^\frown n$, if both are on T). Without loss of generality, we assume v_0 is the root of the tree. If T has an infinite path, then the corresponding graph is robber-win because the robber can move along the infinite path ahead of the cop. Otherwise, T is cop-win because the cop can chase the robber to a leaf. Building on this intuition, Stahl [16] proved that any robber-win strategy on T computes an infinite path. Her argument to code hyperarithmetic sets into robber-win strategies carries over immediately to games with n-cops.

Lemma 2.1 (Stahl [16])**.** *Let $T \subseteq \omega^{<\omega}$ be a tree with an infinite path. Every robber-win strategy computes an infinite path in T.*

Theorem 2.2. *For each computable ordinal α and $n \geq 1$, there is a robber-win graph in the game with n cops such that every robber-win strategy computes $\mathbf{0}^{(\alpha)}$.*

Proof. Let T be a computable tree with infinite paths such that every path computes $\mathbf{0}^{(\alpha)}$. We assume without loss of generality that all the cops start at the root node. Because the graph is a tree, there is a unique shortest path between the cops and the robber and the distance-minimizing strategy of moving directly towards the robber is the optimal cop strategy. It follows that there is no advantage from having n cops and hence Lemma 2.1 (or more precisely, its proof) applies in this context as well. $\square$

Turning to cop strategies, for each $n \geq 2$, we show there is a computable graph that can be won computably by n cops but not by $n-1$ cops, with no limitations on the complexity of the robber strategy. Furthermore, we realize this behavior on graphs that are classically cop-win by a single cop.

Theorem 2.3. *For each $n \geq 2$, there is a computable graph G_n that is cop-win, has a computable n-cop winning strategy, but does not have a computable $(n-1)$-cop winning strategy.*

Proof. Fix $n \geq 2$. We suppress the subscript n, writing G in place of G_n. Initially, G consists of a root node v_0 (where the cops start) attached to infinitely many nodes x^e extended by two node chains as shown in Figure 1.

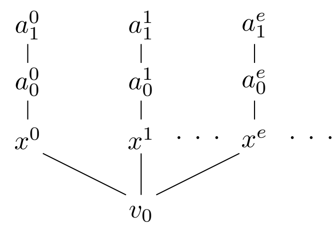

FIGURE 1. Initial set-up for G.

To diagonalize against the partial computable function Φ_e being an $(n-1)$-cop winning strategy, we build a subgraph called the *e-section* starting with the nodes x^e, a_0^e and a_1^e. The e-sections are disjoint and only accessible through x^e, so we construct them independently. We assume the reflexive edges are added automatically and will not explicitly mention them or show them in diagrams.

To decide how to expand the e-section, we simulate a play of the game with $n-1$ cops, allowing Φ_e to control cops. In the simulated game, the robber starts at a_1^e. We do not add points to the e-section unless Φ_e moves a cop into a position adjacent to the robber. At that stage, we expand the e-section to give the robber an escape route in a controlled way so that G remains cop-win and so that n computable cops can win on G. For ease of notation, we fix e and drop the superscripts.

Consider Φ_e on the initial cop position 0^{n-1} (i.e. all start at v_0) and the initial robber position a_1. At each stage s, we check if $\Phi_{e,s}(\langle 0^{n-1}, a_1\rangle)$ converges, and if so, whether it moves a cop to a_0. If $\Phi_{e,s}(\langle 0^{n-1}, a_1\rangle)\downarrow = \gamma_1$ but a_0 is not one of the cop positions in γ_1, then we leave the robber at a_1 in our simulation and at future stages, check $\Phi_{e,s}(\langle 0^{n-1}, a_1, \gamma_1, a_1\rangle)$. We continue this process each time Φ_e converges without placing a cop at a_0 (assuming Φ_e outputs legal moves).

More formally, suppose we have completed the j-th round without the cop positions $\gamma_1, \ldots, \gamma_j$ containing a_0. If we see $\Phi_{e,s}(\langle 0^{n-1}, a_1, \ldots, \gamma_j, a_1\rangle)\downarrow = \gamma_{j+1}$ but a_0 is not one of the cop positions in γ_{j+1}, then we check $\Phi_e(\langle 0^{n-1}, a_1, \ldots, \gamma_{j+1}, a_1\rangle)$ at future stages. We refer to this process as "watching Φ_e while keeping the robber at a_1 until a cop is moved adjacent to a_1."

If Φ_e outputs a non-legal move, then Φ_e has lost already and we can stop building the e-section. Therefore, without loss of generality, we assume whenever Φ_e outputs a tuple γ_{j+1} representing the cop positions, each node $\gamma_{j+1}(k)$ has already been placed in G with an edge to the node $\gamma_j(k)$.

If Φ_e eventually moves a cop to a_0, then we add vertices a_2 and y_i^0, for $i < n$, to the e-section with edges from a_2 to a_1 and from each each y_i^0 to a_0, a_1, a_2 and x. For readability, we show the connections just from y_0^0 in Figure 2. (Note that the y_i^0 nodes are not connected to each other.)

No cop in the simulated game is currently at a neighbor of a_2, because we just added the y_i^0 nodes and because no cop can be at a_1 since the first cop only just arrived at a_0. Therefore, in our simulated game, we can move the robber to a_2 knowing that each cop is at least two nodes away from a_2.

We now enter the main loop of the construction of the e-section. We continue the process described above to compute the movement of the cops under Φ_e, keeping

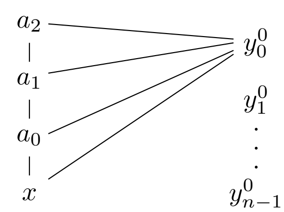

FIGURE 2. Nodes added when a cop moves adjacent to a_1

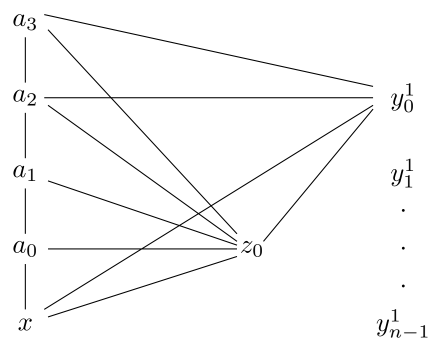

FIGURE 3. Nodes added when a cop moves adjacent to a_2.

the robber at a_2 unless a cop moves to a node adjacent to a_2. If Φ_e never moves a cop adjacent to a_2, then the robber wins and Φ_e is not a winning strategy.

Suppose Φ_e moves a cop to a_1 or some y_i^0. Since Φ_e only controls $n-1$ cops, at least one node y_j^0 does not contain a cop. Let z_0 denote such a node. We expand the e-section by adding vertices a_3 and y_i^1 for $i < n$. We add edges from a_3 to a_2 and z_0, and from each y_i^1 to a_2, a_3, z_0 and x. For readability, we only show the edges from y_0^1 in Figure 3 and we don't show the y_i^0 nodes other than z_0.

We repeat this process. No cop is currently adjacent to a_3 in the simulated game, so we move the robber to a_3 and continue to compute the trajectory of the cops using Φ_e, keeping the robber at a_3 unless a cop moves to an adjacent node.

If a cop moves to z_0, a_2, or some y_i^1, we expand the e-section by adding vertices a_4 and y_i^2 for $i < n$. There is at least one y_j^1 that does not contain a cop and we let z_1 denote such a node. We add edges from a_4 to a_3 and z_1, and from each y_i^2 to a_3, a_4, z_1 and x. In Figure 4, we only show edges from y_0^2 and we don't show the y_i^0 or y_i^1 nodes other than z_0 and z_1.

In general, for $k \geq 1$, when a_{k+2} is added, it is connected to a_{k+1}, z_{k-1} and each y_i^k with $i < n$. We simulate the game using Φ_e with the robber at a_{k+2}. If Φ_e moves a cop to one of these neighbors of a_{k+2}, the graph is expanded by adding a_{k+3} and y_i^{k+1} for $i < n$. We set z_k to be a node of the form y_j^k that does contain have a cop. We add edges from a_{k+3} to a_{k+2} and z_k, and from each y_i^{k+1} to a_{k+3}, a_{k+2}, z_k and x. We move the robber to a_{k+3} and continue the simulated game.

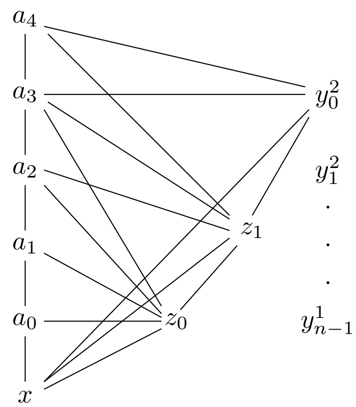

FIGURE 4. Nodes added when a cop moves adjacent to a_3.

This completes the description of the construction. Note that the e-section could be finite or infinite. The nodes z_k are connected to a_{k+1}, a_{k+2}, a_{k+3}, x, y_i^{k-1} and y_i^{k+1} for $i < n$ (except for z_0 which is also connected to a_0, and there are no nodes y_i^{-1}). The other nodes of the form y_i^k are connected to a_{k+1}, a_{k+2}, x and z_{k-1} (except for y_i^0 which is also connected to a_0, and there is no node z_{-1}).

In each e-section, we refer to the set of a_j points as the *vertical segment* and the set of points y_i^j as the *horizontal section*. Importantly, x is connected to every point in the horizontal section.

It remains to check the properties of G stated in the theorem. It is immediate that if Φ_e is a computable $(n-1)$-cop strategy, then the robber can beat Φ_e by starting at the node a_1^e and moving up the vertical segment of the e-section each time a new node a_{k+1}^e appears (i.e. when Φ_e moves a cop adjacent to the robber).

Lemma 2.4. *The graph G can be won with n cops following a computable strategy.*

Proof. The cops start at v_0, so we can assume the robber does not. Fix e such that the robber starts in the e-section. In the first round, all the cops move to x^e. The remainder of the game will stay in the e-section, so we drop the superscripts. If the robber's next move is to the horizontal segment or to a_0, then a cop can move directly there and win. Therefore, assume the robber moves to a_k with $k \geq 1$.

If $k \geq 2$, then the nodes y_i^{k-2} for $i < n$ are already in G. The cops move to place one cop on each y_i^{k-2}, so one of the cops is on z_{k-2}. The nodes connected to a_k (if they exist in G) are a_{k-1}, a_k, a_{k+1}, z_{k-3}, and y_i^{k-2} and y_i^{k-1} for $i < n$. There are cops already on each y_i^{k-2}. The nodes connected to z_{k-2} (if they exist in G) are a_{k-1}, a_k, a_{k+1}, and y_i^{k-3} (which includes z_{k-3}) and y_i^{k-1} for $i < n$. Therefore, z_{k-2} covers every other node the robber could move to.

If $k = 1$ and the nodes y_i^0 have appeared in G, then the cops move to these nodes and win as above. However, these nodes might not have appeared in G when the cop has to move (and might never be added). In this case, the cops all move to a_0. If the robber remains at a_1, then he loses on the next turn. Suppose the nodes a_2 and y_i^0 appear in G after the cops move to a_0. If the robber moves to any node y_i^0, he loses on the next turn. Therefore, his only choice is to move to a_2. The cops can then move from a_0 to the nodes y_i^0 and proceed as above to win. □

Lemma 2.5. *The graph G can be won by 1 cop following a classical strategy.*

Proof. The cop starts at v_0. We assume the robber starts in the e-section of the graph and the cop moves to x^e. As above, we assume the robber moves to a_k with $k \geq 1$. The key difference in this lemma is that the classical cop knows the entire graph at the end of the construction rather than having to respond at a finite stage when the graph is not yet complete. The proof breaks into five cases.

Suppose the robber is at a_1 and there is no node a_2 in G. The cop moves to a_0 and then to a_1 to capture the robber.

Suppose the robber is at a_1 or a_2 and there is no node a_3 in G. The cop moves to a_0. The only viable option for the robber is to move to (or stay on) a_2. The cop now moves to a_1, which covers all the nodes connected to a_2.

Suppose the robber is at a_k for $k \geq 3$ and there is no node a_{k+1} in G. The cop moves to y_0^{k-2}. The only viable option for the robber is to move to y_i^{k-2} for some $0 < i < n$. The cop can move to z_{k-3} which covers all the nodes connected to y_i^{k-2}.

Suppose the robber is at a_k with $k \geq 2$ and a_{k+1} exists in G. The cop moves to z_{k-2}. As above, the only nodes connected to a_k that are not connected to z_{k-2} are the $y_i^{(k-2)}$ (other than z_{k-2}). However, if the robber moves to $y_i^{(k-2)}$, the cop can move to z_{k-3}. The only nodes connected to $y_i^{(k-2)}$ (which, recall, is not z_{k-2}) are x, z_{k-3}, a_{k-1} and a_k. However, each of these is connected to z_{k-3}, so the cop can win on her next turn.

Finally, suppose the robber is at a_1 and a_3 exists. In this case, the cop can move to z_0. Again, the only nodes connected to a_1 that are not connected to z_0 are the y_i^0 (other than z_0). However, if the robber moves to y_i^0, then the cop can move to a_0 which covers all the nodes connected to y_i^0. □

□

3. Index sets

In this section, we show that for each $n \geq 2$, the index set of computable n-cop-win graphs is Π^1_1-complete. The proof resembles the proof in Stahl [16] that the index set of computable cop-win graphs is Π^1_1 complete. The main difference is that we use a characterization of n-cop-win graphs by Clarke and MacGillivray [8]. Clarke and MacGillivray prove the characterization for finite graphs, although they note it can be modified for infinite graphs. For completeness, we make these modifications and give a proof that works for all graphs.

Let G be a graph with vertex set V and edge relation E. For $v \in V$, we let $N_G[v] = \{x \in V : Evx \text{ holds}\}$ denote the set of neighbors of v. The n-th *categorical product* G^n of G is the graph with vertex set V^n and an edge between $p, q \in V^n$ if and only if for all $i < n$, $Ep(i)q(i)$ holds. If G is connected and reflexive, then so is the product graph G^n. Fixing notation, throughout this section, we use v, x and y to range over V and p and q to range over V^n.

To keep track of the cop positions in the n-cop game, we imagine a single player moving on G^n rather than n different cops moving on G. We define a family of relations $\leq_\alpha$ on $V \times V^n$ indexed by ordinals. The intuition for $v \leq_\alpha p$ is that the cops can win in α turns when the robber is on v, the cops are on $p(0), \ldots, p(n-1)$ and it is the robber's turn to move.

Formally, the relations $\leq_\alpha$ are defined by transfinite recursion. For $v \in V$ and $p \in V^n$, $v \leq_0 p$ if and only if $v = p(i)$ for some $i < n$. That is, $v \leq_0 p$ indicates that the cops have captured the robber. For $\alpha > 0$, $v \leq_\alpha p$ if and only if

$$(\forall y \in N_G[v])(\exists q \in N_{G^n}[p])(\exists \beta < \alpha)\,(y \leq_\beta q) \tag{1}$$

In other words, no matter where the robber moves on his next turn, the cops can improve their positions relative to the robber when they move.

It follows immediately from this definition that $\gamma \leq \alpha$ implies $\leq_\gamma \subseteq \leq_\alpha$. Therefore, there is an ordinal δ at which this sequence stabilizes (i.e. at which $\leq_\alpha = \leq_\delta$ for all $\alpha \geq \delta$). We let $\preceq_G$ denote this limiting relation $\leq_\delta$. We prove Clarke and MacGillivray's theorem with no restrictions on the size of the graph.

Theorem 3.1 (Clarke and MacGillivray [8])**.** *G is n-cop-win if and only if the relation $\preceq_G$ is trivial (i.e. $v \preceq_G p$ for all $v \in V$ and $p \in V^n$).*

Proof. Suppose $\preceq_G$ is trivial. Let $p_0 = \langle v_0, \ldots, v_0 \rangle$ be the cops' fixed starting position, let x_0 be the robber's starting position, and assume $p_1 = p_0$ (i.e. the cops don't move on their first turn). Since $\preceq_G$ is trivial, we can fix an ordinal α_0 such that $x_0 \leq_{\alpha_0} p_1$. In general, let x_k denote the position of the robber in round k, p_{k+1} denote the positions of the cops in round $k+1$, and α_k be the least ordinal such that $x_k \leq_{\alpha_k} p_{k+1}$. By condition (1), for any robber move x_{k+1}, the cops can choose p_{k+2} so that $x_{k+1} \leq_{\alpha_{k+1}} p_{k+2}$ with $\alpha_{k+1} < \alpha_k$. Therefore, in finitely many rounds, we arrive at $x_\ell \leq_0 p_{\ell+1}$ and the cops win.

Conversely, suppose that $\preceq_G$ is not trivial. Fix α such that $\preceq_G = \leq_\alpha = \leq_{\alpha+1}$. Consider any $x \in V$ and $p \in V^n$ such that $x \not\preceq_G p$. Since $x \not\leq_{\alpha+1} p$, condition (1) fails, and so there is a node y adjacent to x such that $y \not\leq_\alpha q$ (and hence $y \not\preceq_G q$) for all cop positions q adjacent to p. It follows that if the players are at positions such that $x \not\preceq p$, then the robber can move to maintain this condition and thus avoid capture forever. In particular, the robber has a strategy to win if the cops start at the positions specified by p. However, since the starting position for the cops doesn't affect whether G is n-cop win or not, we conclude that G is robber-win in the game with n cops. □

For a total computable function Φ_e, we define the computable graph G_e to have vertex set ω with Φ_e giving the characteristic function of the edge relation: if $\Phi_e(x, y) = 1$, then there an edge between x and y, and if $\Phi_e(x, y) = 0$, then there is no edge. It is arithmetical (even Π^0_2) to say Φ_e is total and the resulting computable graph G_e is reflexive and connected. We form the computable product graph G_e^n in the obvious way by considering computations on n-tuples.

Theorem 3.2. *For each $n \geq 2$, the index set $I_n = \{e : G_e$ is a n-cop-win graph$\}$ is Π^1_1-complete.*

Proof. We have already noted that a tree $T \subseteq \omega^{<\omega}$ is n-cop-win if and only if T has no infinite path. Since the relation of a computable tree having no infinite path is Π^1_1-hard, it follows that I_n is Π^1_1-hard. Therefore, it suffices to show that I_n has a Π^1_1 definition.

Our Π^1_1 definition will say that $e \in I_n$ if and only if G_e is a computable reflexive connected graph and every relation that "looks like" $\preceq_{G_e}$ is trivial. The first condition is arithmetical, so we focus on the second condition.

Suppose G_e is a computable reflexive connected graph. The relation $\preceq_{G_e}$ has two fundamental properties. First, since $\leq_0 \subseteq \preceq_{G_e}$, we have $x \preceq_{G_e} p$ whenever $x = p(i)$ for some $i < n$. Second, by condition (1), if $(\forall y \in N_G[x])(\exists q \in N_{G^n}[p])\,(y \preceq_{G_e} q)$, then $x \preceq_{G_e} p$.

We capture these properties by arithmetic formulas with a free variable R ranging over $(n+1)$-arity relations. In these formulas, let x and y be single variables and let p and q be n-tuples of variables $p_0, \dots, p_{n-1}$ and $q_0, \dots, q_{n-1}$ (i.e. x, y range over the vertices in G_e and p, q range over the vertices in G_e^n). Let $\psi_1(R)$ capture the first property

$$(\forall x, p)\left((x = p_0 \vee \cdots \vee x = p_{n-1}) \to R(x,p)\right)$$

and let $\psi_2(R)$ capture the second property

$$(\forall x, p)\Big[(\forall y \in N_G[x])(\exists q \in N_{G_e^n}[p])(R(y,q)) \to R(x,p)\Big].$$

We claim that G_e is a n-cop-win if and only if G_e is a computable reflexive connected graph such that

$$(\forall R \subseteq G_e \times G_e^n)\left((\psi_1(R) \wedge \psi_2(R)) \to (\forall x, p) R(x,p)\right)$$

Since this formula is Π^1_1, verifying this claim will complete the proof.

Suppose the offset Π^1_1 formula holds for G_e. Since ψ_1 and ψ_2 hold for the relation $\preceq_{G_e}$, it follows that $x \preceq_{G_e} p$ for all x and p. Therefore, by Clarke and MacGillivray's theorem, G_e is k-cop-win.

Suppose G_e is k-cop-win and R is a relation such that $\psi_1(R)$ and $\psi_2(R)$ hold. Since $\psi_1(R)$ holds, we have $\leq_0 \subseteq R$. Since $\psi_2(R)$ holds, it follows that if $\leq_\beta \subseteq R$ for all $\beta < \alpha$, then $\leq_\alpha \subseteq R$. Consequently, $\leq_\alpha \subseteq R$ for all α, and hence $\preceq_{G_e} \subseteq R$. The relation $\preceq_{G_e}$ is trivial by Clarke and MacGillivray's theorem, and therefore, $R(x,p)$ holds for all x and p as required. □

References

[1] M. Aigner and M. Fromme, "A game of cops and robbers," *Discrete Applied Mathematics* **8**, 1984, 1-11.

[2] C.J. Ash and J.F. Knight, *Computable Structures and the Hyperarithmetic Hierarchy*, Elsevier, 2000.

[3] W. Baird, A. Bonato, A. Beveridge, P. Codenotti, A. Maurer, J. McCauley and S. Valeva, "On the minimum order of k-cop-win graphs," *Contributions to Discrete Mathematics*, 2014, 70-84.

[4] A. Bonato, *An Invitation to Pursuit-Evasion Games and Graph Theory*, American Mathematical Society, 2022.

[5] A. Bonato and R.J. Nowakowski, *The Game of Cops and Robbers on Graphs*, American Mathematical Society, 2011.

[6] M. Chastand, F. Laviolette and N. Polat, "On constructible graphs, infinite bridged graphs and weakly cop-win graphs," *Discrete Mathematics* **224**, 2000, 61-78.

[7] N.E. Clarke, *Constrained Cops and Robbers*, Ph.D. Thesis, Dalhousie University, 1999.

[8] N.E. Clarke and G. MacGillivray, "Characterizations of k-cop win graphs," *Discrete Math* **312**, 2012, 1421-1425.

[9] L. Evron, R. Solomon and R.D. Stahl, "Dominating ordersm vertex pursuit games and computability theory," submitted.

[10] S.A. Hosseini, "A note on k-cop-win graphs," *Discrete Math* **341**, 2018, 1136-1137.

[11] M-R. Ivan, I. Leader and M. Walters, "Constructible graphs and pursuit," *Theoretical Computer Science* **930**, 2022, 196-208.

[12] F. Lehner, "Pursuit evasion on infinite graphs," *Theoretical Computer Science* **655** (Part A), 2016, 30-40.

[13] R.J. Nowakowski and P. Winkler, "Vertex-to-vertex pursuit in a graph," *Discrete Mathematics* **43**, 1983, 235-239.
[14] A. Quilliot, "Jeux et pointes fixes sur les graphes," *Thèses de 3ème cycle*, Université de Paris VI, 1978, 131-145.
[15] R.I. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag, 1987.
[16] R.D. Stahl, "Computability and the game of cops and robbers on graphs," *Archive for Mathematical Logic* **61**, 2022, 373-397.

Department of Mathematics, University of Connecticut, Storrs, CT 06269
Email address: `alexa.mcleod@uconn.edu`
Email address: `david.solomon@uconn.edu`