

A simple method of computing the catch time

Nancy E. Clarke *

*Department of Mathematics and Statistics, Acadia University
Wolfville, NS, Canada B4P 2R6*

Stephen Finbow *

*Department of Mathematics, Statistics and Computer Science
St. Francis Xavier University
P.O. Box 5000, Antigonish, NS, Canada B2G 5W3*

Gary MacGillivray *

*Department of Mathematics and Statistics, University of Victoria
P.O. Box 3060 STN CSC, Victoria, BC, Canada V8W 3R4*

Received 17 February 2012, accepted 9 May 2013, published online 6 September 2013

Abstract

We describe a simple method for computing the maximum length of the game cop and robber, assuming optimal play for both sides.

Keywords: Pursuit game, Cops and Robber, catch time.

Math. Subj. Class.: 05C57, 91A43

1 Introduction

The perfect information pursuit game “Cop and Robber” was introduced independently by Quilliot [13] in 1978, and Nowakowski and Winkler in 1983 [12]. It is played on a graph by two sides called *the cop* and *the robber*. The cop begins the game by choosing a vertex to occupy. The robber then chooses a vertex and the two sides move alternately, with the cop moving first. A move for the cop consists of either traversing an edge to a neighbouring vertex, or passing on his turn and remaining at the same vertex. A move for the robber is defined analogously. The cop wins if he *catches* the robber by occupying the same vertex as the robber after a finite number of moves. Otherwise the robber wins. The graphs on

*Research supported by NSERC.

E-mail addresses: nancy.clarke@acadiau.ca (Nancy E. Clarke), sfinbow@stfx.ca (Stephen Finbow), gmacgill@math.uvic.ca (Gary MacGillivray)

which the cop has a winning strategy are called *cop-win*, or *dismantlable*. The book by Bonato and Nowakowski [3] is a wonderful introduction to this game and its variants.

Let G be a cop-win graph. We define the *catch time* of G , first introduced in [2], denoted $\text{catchtime}(G)$, to be the minimum number of cop moves needed to be guaranteed to catch the robber, where the minimum is taken over all possible strategies on the assumption of optimal play by both sides. Passes count as moves. A variety of results imply ways to compute $\text{catchtime}(G)$, and discover the associated optimal strategies, in cubic time using some auxiliary structure [1, 8, 11, 12]. In this paper we describe a method for determining the catch time using only local information about neighbourhoods. It leads to a simple algorithm which is easy to carry out with pencil and paper if a drawing of the graph is available, and which can be implemented to run in cubic time.

2 Cop-win orderings and an associated strategy

There are at least three different characterizations of finite cop-win graphs [8, 12, 13] (also see [7, 10, 11]). The one that is of primary interest here is due to Nowakowski and Winkler, and Quilliot, independently.

Theorem 2.1. [12, 13] *Let G be a finite graph. Then G is cop-win if and only if there exists an enumeration v_1, v_2, \dots, v_n of the vertices of G such that for $i = 1, 2, \dots, n - 1$ there exists $j_i > i$ such that $(N[v_i] \cap \{v_i, v_{i+1}, \dots, v_n\}) \subseteq N[v_{j_i}]$.*

The notation $N[x]$ that appears in the above theorem and elsewhere denotes the *closed neighbourhood* of x , defined to be the set containing x and all vertices adjacent to x .

The enumeration of the vertices of G in Theorem 2.1 has come to be known as a *cop-win ordering* or *dismantling ordering* of G . A vertex $x \in V(G)$ is called a *corner* if there exists $y \in V(G) - \{x\}$ such that $N[y] \supseteq N[x]$, or if $V(G) = \{x\}$. When such a vertex y exists, we say that it *covers* x . Cop-win orderings are constructed by iteratively deleting corners from the graph (also, see [5, 6]). We will make use of the following propositions which are key to the proof of Theorem 2.1 and the associated strategy. In Section 4 we will show how our results can be viewed as a generalization of this theorem. The methods of proof we use are essentially those of Nowakowski and Winkler, and Clarke [7, 9, 10, 12].

Proposition 2.2. [12, 13] *Let x be a corner of the graph G . Then G is cop-win if and only if $G - x$ is cop-win.*

By *optimal play* we mean that the cop seeks to catch the robber as quickly as possible, and the robber seeks to evade capture for as long as possible. Various aspects of optimal play are considered by Boyer *et al.* [4]. They show that in some cases of optimal play it is necessary for the cop to revisit a vertex, and in some cases it is necessary for the distance between the cop and robber to increase at some time.

Proposition 2.3. [12, 13] *If G is cop-win then, assuming optimal play, just before the robber's last move, the robber is on a corner of G and the cop is on a vertex that covers it. The game ends on the cop's next move.*

We now give a description of the cop's strategy that arises from the proof of Theorem 2.1. The following is applied recursively for $i = 1, 2, \dots, n - 1$. The vertex v_i is a corner of G_i , the subgraph of G induced by $\{v_i, v_{i+1}, \dots, v_n\}$. (Note that $G_1 = G$.) While the robber plays the game on G_i , the cop plays as if it were on $G_{i+1} = G_i - v_i$, that is, if

the robber moves to v_i the cop plays his winning strategy for G_{i+1} as if the robber were on a particular vertex y that covers v_i . By Proposition 2.2, the cop has a winning strategy on G_{i+1} . Thus, at some point either the cop and robber occupy the same vertex, or the cop is on y and the robber is on v_i . In the former case the game on G_i is also over, and in the latter case it is over after one more cop move. It can be proved by induction that this strategy leads to the bound $catchtime(G) \leq |V| - 1$. Essentially the same argument is used to prove Theorem 3.3. The strategy can be formulated using retractions [7, 10]. We do the same for our results in Section 4.

3 Finding the catch time

Cop-win orderings are constructed by deleting corners from the graph one at a time. The key to the method described below for finding the catch time is deleting many corners simultaneously. Since it is possible for every vertex of a graph (cop-win or not) to be a corner – informally, for any graph G replace each vertex by a copy of K_2 and add all possible edges between copies of K_2 that replaced adjacent vertices of G – it is necessary to have a means of selecting which subset of the corners to delete.

Let G be a graph. Define an equivalence relation Θ_G on V by $(u, w) \in \Theta_G$ if and only if $N[u] = N[w]$. Note that the subgraph of G induced by each equivalence class is complete, and either there are no edges joining vertices in different equivalence classes, or all possible edges joining them are present. Let G / Θ_G be the graph whose vertices are the equivalence classes $\{[x] : x \in V\}$ of Θ_G , with $[x]$ adjacent to $[y]$, where $y \notin [x]$, if and only if $xy \in E(G)$ (that is, every vertex in $[x]$ is adjacent to every vertex in $[y]$). Equivalently, G / Θ_G is the subgraph of G induced by selecting one vertex from each equivalence class of Θ_G . By construction, no two vertices of G / Θ_G have the same neighbourhood. Thus, if $[y]$ covers $[x]$ in G / Θ_G , the neighbourhood of $[y]$ properly contains the neighbourhood of $[x]$. Hence, if G is not complete, every vertex x belonging to a corner of G / Θ_G is covered in G by a vertex y such that $N_G[y] \supset N_G[x]$. In particular, for every such x there exists such a y for which $[y]$ is not a corner of G / Θ_G . The main purpose of introducing the equivalence relation Θ_G is to assure that “twins”, that is, vertices with identical neighbourhoods, are treated in exactly the same way.

Proposition 3.1. *Let G be a graph which is not complete and $X \subseteq V$ be the set of vertices belonging to corners of G / Θ_G . Then G is cop-win if and only if $G - X$ is cop-win.*

Proof. Let $X = \{x_1, x_2, \dots, x_k\}$. Since G is not complete, X is a proper subset of V . Since each vertex in X is covered by a vertex in the non-empty set in $V - X$, we have, by Proposition 2.2, that G is cop-win if and only if $G - x_1$ is cop-win if and only if $G - \{x_1, x_2\}$ is cop-win, and so on until, finally, if and only if $G - X$ is cop-win. \square

Let G be a cop-win graph. By Proposition 3.1 we can define an ordered partition X_1, X_2, \dots, X_k of V by setting $G_i = G - \cup_{t=1}^{i-1} X_t$ (so that $G_1 = G$), and X_i to be the set of corners of G_i / Θ_{G_i} . Note that the subgraph of G induced by the top layer X_k is necessarily a complete graph. We call X_1, X_2, \dots, X_k the *cop-win partition of G* and, for $i = 1, 2, \dots, k$ call the set X_i the i -th layer. Note that X_i, X_{i+1}, \dots, X_k is the cop-win partition of G_i .

Proposition 3.2. *A graph is cop-win if and only if it has a cop-win partition.*

Proof. By the discussion above we need only show that a graph with a cop-win partition is cop-win. Suppose that X_1, X_2, \dots, X_k of G is a cop-win partition of G . The enumeration of $V(G)$ constructed by first listing the vertices in X_1 (in any order), then those in X_2 , and so on until, finally, the vertices in X_k are listed is a cop-win ordering. The result now follows from Theorem 2.1. \square

Suppose that G has a cop-win partition X_1, X_2, \dots, X_k , where $k > 1$. If some vertex of X_k is adjacent to all vertices of X_{k-1} , then all vertices of X_k must be adjacent to all vertices of X_{k-1} : since the subgraph induced by X_k is complete, any vertex of X_k with a non-neighbour in X_{k-1} would belong to a corner in $G_{k-1} / \Theta_{G_{k-1}}$. Further, if $X_k = \{x_k\}$ then x_k is adjacent to every vertex of X_{k-1} as, in G_{k-1} , any such vertex must be covered by a vertex in X_k .

Theorem 3.3. *Let G have a cop-win partition X_1, X_2, \dots, X_k . Then $\text{catchtime}(G) = k - 1$ if every vertex of X_k is adjacent to every vertex of X_{k-1} , or G has only one vertex. Otherwise, $\text{catchtime}(G) = k$.*

Proof. We first show by induction on k that the robber can be caught in at most the given number of cop moves. If $k = 1$, then G is complete and $\text{catchtime}(G) = 0$ when G has one vertex and $\text{catchtime}(G) = 1$ otherwise. Suppose $k = 2$. If every vertex in X_2 is adjacent to every vertex of X_1 , then an optimal play game will end in one cop move. Otherwise, every vertex in X_2 has a non-neighbour in X_1 . The cop begins by choosing a vertex $x_2 \in X_2$. No matter which vertex non-adjacent to x_2 the robber chooses, by definition of X_1 and since the subgraph induced by X_2 is complete, the cop can move to a vertex of X_2 that covers the robber position. The game ends in one more cop move, as required.

Suppose that the statement holds for all cop-win graphs in which the cop-win partition has $k - 1$ layers. Let G be a cop-win graph for which the cop-win partition has k layers. By Proposition 3.1, the graph $G_2 = G - X_1$ is cop-win and has a cop-win partition with $k - 1$ layers. As before, while the robber plays the game on G , the cop plays the game as if it were on G_2 . If the robber is in X_1 then by definition of X_1 he has a *shadow* (a particular vertex that covers his position) in G_2 . By the induction hypothesis and assuming optimal play, after at most $\text{catchtime}(G_2) - 1$ cop moves, the robber or his shadow is on a vertex $x_2 \in X_2$ and the cop is on a vertex y of G_2 that covers it (the robber could actually be located in X_1 which is covered by x_2). The robber can evade capture for at most two more cop moves. On the next cop move the cop catches the robber's shadow (or possibly the robber) on some vertex z of G_2 . Suppose he has only caught the shadow. Then the robber is on a vertex $x_1 \in X_1$ which is covered by z . No matter to which vertex the robber moves, the cop now has a move to the same vertex, so the game ends on the next cop move.

The proof that there is an optimal play game requiring the given number of moves is also by induction on k . The statement is easy to see when $k = 1$. Suppose it holds for all cop-win graphs in which the cop-win partition has $k - 1$ layers. Let G be a cop-win graph for which the cop-win partition has k layers. By the induction hypothesis and assuming optimal play, there is a game on G_2 that requires $\text{catchtime}(G_2)$ moves. Since there is never an advantage to the cop in using a vertex in X_1 – a cover in G_2 could be used instead – in order to make the game on G last as long as possible, the robber first plays his optimal game on G_2 . By Proposition 2.3, just before the robber's last move in this game, the robber is on a vertex $x_2 \in X_2$ and the cop is on a vertex y that covers it (in G_2). By definition of

X_2 , in the game on G , the robber has a move to a vertex in $x_1 \in X_1$ which is not adjacent to y . Since x_1 is a corner of G , the cop has a move to a vertex that covers x_1 . (A cover of x_1 is adjacent to all neighbours of x_1 , hence is adjacent to x_2 . Since y covers x_2 , it is adjacent to all neighbours of x_2 , including the vertex that covers x_1 .) The game therefore lasts one move longer than before. This completes the proof. \square

Theorem 3.3 implies a straightforward algorithm for computing the catch time. Given a graph G with n vertices, the first step is to find the quotient graph, G / Θ_G . If G is represented by an adjacency matrix, A , then viewing each row as characteristic vector of the corresponding vertex, two vertices belong to the same equivalence class if the corresponding rows are identical. The equivalence classes, and consequently G / Θ_G (choose one vertex from each equivalence class) can be found in $O(n^2)$ time. The corners of G / Θ_G are also easy to find from the rows of A in time $O(n^2)$. Hence the first set, X_1 , in a cop-win partition can be determined in quadratic time. After marking the vertices in X_1 as deleted, the process is repeated with $G - X_1$ and so on, until the cop-win partition X_1, X_2, \dots, X_k is determined. Since $k \leq n$, this process takes time $O(n^3)$. The last step is to test whether every vertex of X_k is adjacent in G to every vertex in X_{k-1} , which takes time $O(n^2)$, and then apply the theorem. Hence the catch time can be determined in cubic time.

We conclude this section by noting that, by definition of the catch time of the cop-win graph G , for any initial position chosen by the cop there is an initial position available to the robber such that optimal play by both sides results in a game of length at least $catchtime(G)$. Further, there exists at least one initial position available to the cop (for example, any one in the “top” layer of the cop-win partition, plus possibly some others) such that optimal play by both sides results in a game of length exactly $catchtime(G)$.

4 Retractions and a description of the strategy

The purpose of this section is to illustrate how our results can be seen as a generalization of Theorem 2.1, and to present the cop’s strategy implied by Theorem 3.3 in the language of retractions (see [7, 10]), thus perhaps making the implied recursion more transparent.

We first describe an alternate approach that could have been used instead of proceeding directly to the cop-win partition as in Section 3. An advantage of using this point of view is that the connection with Theorem 2.1 is immediate.

By a *cop-win layering* of a graph G we mean an ordered partition X_1, X_2, \dots, X_ℓ of $V(G)$ such that for $i = 1, 2, \dots, \ell - 1$ the set X_i is a set of corners of $G_i = G - \cup_{t=1}^{i-1} X_t$ each of which has a cover in G_{i+1} . Each set X_i is called a *layer*.

A cop-win ordering v_1, v_2, \dots, v_n gives a cop-win layering by setting $X_i = \{v_i\}$, $1 \leq i \leq n$. Conversely, a cop-win layering X_1, X_2, \dots, X_k of G gives a cop-win ordering in the same way as described before. The following statements hold using essentially the same proofs as before.

Theorem 4.1. *A graph G is cop-win if and only if it admits a cop-win layering.*

Theorem 4.2. *If G has a cop-win layering X_1, X_2, \dots, X_k , then $catchtime(G) \leq k - 1$ if every vertex of X_k is adjacent to every vertex of X_{k-1} , or G has only one vertex. Otherwise, $catchtime(G) \leq k$.*

Let $\mathcal{L} = X_1, X_2, \dots, X_k$ be a cop-win layering of G . Define $\mu_G(\mathcal{L}) = k - 1$ if every vertex of X_k is adjacent to every vertex of X_{k-1} or if G has only one vertex, and $\mu_G(\mathcal{L}) = k$ otherwise.

Theorem 4.3. *The maximum length of the cop and robber game on the cop-win graph G equals $\min_{\mathcal{L}} \mu_G(\mathcal{L})$, where the minimum is over all cop-win layerings of G .*

Given a cop-win ordering v_1, v_2, \dots, v_n of G , Theorem 4.2 can be used to improve the upper bound of $n - 1$ cop moves in the associated strategy. Define a cop-win layering by letting X_1 be a maximum size set of consecutive vertices $X_1 = \{v_1, v_2, \dots, v_{i_1}\}$ such that each vertex in X_1 has a cover in $V(G) - X_1$. Now delete the vertices in X_1 and define X_2 in the same way using the graph $G_2 = G - X_1$. Continue in this way until, finally, a cop-win layering $\mathcal{L} = X_1, X_2, \dots, X_\ell$ is defined. Then $\text{catchtime}(G) \leq \mu_G(\mathcal{L})$.

For the purposes of what follows, it is convenient to regard the graphs under consideration as being reflexive, that is, having a loop at each vertex. A pass corresponds to moving along the loop from a vertex to itself.

Let G be a graph and H be a fixed subgraph of G . A *retraction of G to H* is a homomorphism of G to H that maps H identically to itself. Formally, it is a function $f : V(G) \rightarrow V(H)$ such that $f(h) = h$ for all vertices h of H , and if $xy \in E(G)$ then $f(x)f(y) \in E(H)$. If there exists a retraction of G to H , then H is called a *retract of G* .

Theorem 4.4. [12, 14] *Any retract of a cop-win graph is cop-win.*

Retractions provide a convenient way of describing the cop's strategy arising from Theorem 2.1 [7, 10] (also see [9]). The same method can be used to describe the strategy arising from Theorem 4.2. Suppose that G has at least two vertices and let $\mathcal{L} = X_1, X_2, \dots, X_\ell$ be a cop-win layering of the vertices of G . For $i = 1, 2, \dots, \ell - 1$ there is a retraction f_i of the graph $G_i = G - \cup_{t=1}^{i-1} X_t$ to G_{i+1} that maps each vertex in X_i to a vertex in G_{i+1} that covers it. (If there is more than one candidate for this vertex, it does not matter which one is chosen.) When G is a complete graph with at least two vertices, or every vertex in X_ℓ is non-adjacent to some vertex in $X_{\ell-1}$ there is also a retraction f_ℓ of the (complete) subgraph induced by X_ℓ to any one of its vertices. On his j -th move, the cop plays on $G_{\ell-j+1} = f_{\ell-j} \circ \dots \circ f_2 \circ f_1(G) = G - \cup_{t=1}^{\ell-j} X_t$. We allow $j = 0$ when $|X_\ell| > 1$ and every vertex in X_ℓ has a non-neighbour in $X_{\ell-1}$. No matter where the robber is located in G , his *shadow* (i.e. image under the mapping $f_{\ell-j} \circ \dots \circ f_2 \circ f_1$) is located on one of the vertices of this graph. Since the cop is "on" the robber's shadow after making his move when $j = 1$, and the retractions allow him to stay on it in each subsequent move, the cop is guaranteed to catch the robber after at most $\mu_G(\mathcal{L})$ moves.

References

- [1] A. Berarducci and B. Intriglia, On the cop number of a graph, *Advances in Applied Math.* **14** (1993), 389–403.
- [2] A. Bonato, P. A. Golovach, G. Hahn and J. Kratochvíl, The Capture Time of a Graph, *Discrete Math.* **309** (2009), 5588–5595.
- [3] A. Bonato and R. J. Nowakowski, *The game of Cops and Robbers on Graphs*, AMS Student Mathematical Library, AMS, Providence, RI, 2011.
- [4] M. Boyer, S. El Harti, A. El Ouarari, R. Ganian, T. Gavenčiak, G. Hahn, C. Moldenauer, I. Rutter, B. Thériault, M. Vatschelle, Cops-and-robbers: remarks and problems, manuscript, 2011.
- [5] V. Chepoi, Bridged graphs are cop-win graphs: an algorithmic proof, *J. Combin. Theory Ser. B* **69** (1997), 97–100.

- [6] V. Chepoi, On Distance-preserving and Domination Elimination Orderings, *SIAM J. Discrete Math.* **11** (1998), 414–436.
- [7] N. E. Clarke, *Constrained Cops and Robber*, Ph.D. Thesis, Dalhousie University, 2002.
- [8] N. E. Clarke and G. MacGillivray, Characterizations of k -copwin graphs, *Discrete Math.* **312** (2012), 1421–1425.
- [9] N. E. Clarke and R. J. Nowakowski, Cops, Robber and Photo Radar, *Ars Combinatoria* **56** (2000), 97–103.
- [10] N. E. Clarke and R. J. Nowakowski, Cops, Robber, and Traps, *Utilitas Mathematica* **60** (2001), 91–98.
- [11] G. Hahn and G. MacGillivray, An algorithmic characterisation of k -copwin graphs and digraphs, *Discrete Math.* **306**, 2492–2497.
- [12] R. J. Nowakowski and P. Winkler, Vertex to Vertex Pursuit in a Graph, *Discrete Math.* **43** (1983), 23–29.
- [13] A. Quilliot, *Jeux de Points Fixes sur les graphes*, Thèse de 3ème cycle, Université de Paris VI, 1978, 131–145.
- [14] A. Quilliot, *Problèmes de jeux, de point Fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes*, Thèse d’Etat, Université de Paris VI, 1983.