

Pursuit-Evasion in Graphs: Zombies, Lazy Zombies and a Survivor

Prosenjit Bose ✉ 🏠

School of Computer Science, Carleton University, Ottawa, Canada

Jean-Lou De Carufel ✉

School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Canada

Thomas Shermer ✉

School of Computing Science, Simon Fraser University, Burnaby, Canada

Abstract

We study *zombies and survivor*, a variant of the game of cops and robber on graphs. In this variant, the single survivor plays the role of the robber and attempts to escape from the zombies that play the role of the cops. The zombies are restricted, on their turn, to always follow an edge of a shortest path towards the survivor. Let $z(G)$ be the smallest number of zombies required to catch the survivor on a graph G with n vertices. We show that there exist outerplanar graphs and visibility graphs of simple polygons such that $z(G) = \Theta(n)$. We also show that there exist maximum-degree-3 outerplanar graphs such that $z(G) = \Omega(n/\log(n))$.

Let $z_L(G)$ be the smallest number of *lazy zombies* (zombies that can stay still on their turn) required to catch the survivor on a graph G . We show that lazy zombies are more powerful than normal zombies but less powerful than cops. We prove that $z_L(G) \leq 2$ for connected outerplanar graphs and this bound is tight in the worst case. We show that $z_L(G) \leq k$ for connected graphs with treedepth k . This result implies that $z_L(G)$ is at most $(k+1)\log n$ for connected graphs with treewidth k , $O(\sqrt{n})$ for connected planar graphs, $O(\sqrt{gn})$ for connected graphs with genus g and $O(h\sqrt{hn})$ for connected graphs with any excluded h -vertex minor. Our results on lazy zombies still hold when an adversary chooses the initial positions of the zombies.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Pursuit-evasion games, Outerplanar, Graphs, Treedepth, Treewidth

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2022.56

Related Version *Full Version*: <https://arxiv.org/abs/2204.11926>

Funding Research supported in part by NSERC.

Acknowledgements P. Bose would like to thank P. Morin for fruitful discussions on various aspects of structural graph theory and for bringing reference [15] to his attention.

1 Introduction

The game of cops and robber was first introduced by Quillot in his doctoral thesis [21] and then independently by Nowakowski and Winkler [20]. In this pursuit-evasion game, a set of cops move along the edges of a connected and undirected graph G to catch a robber that is also moving along the edges of G . At the beginning, each cop chooses a starting vertex (multiple cops can occupy the same vertex). Then the robber chooses a starting vertex. From there, when it is the cops' turn to play, each cop decides either to stay still or to move to an adjacent vertex. When it is the robber's turn to play, the robber decides either to stay still or to move to an adjacent vertex. (In the classical version of the game, the cops and the robber are aware of each others' locations at all times.) If at least one cop shares a vertex with the robber, then the cops win. However, if the robber indefinitely avoids capture, then the robber wins. The *cop number* of a graph G , denoted as $c(G)$, is the minimum



© Prosenjit Bose, Jean-Lou De Carufel, and Thomas Shermer;

licensed under Creative Commons License CC-BY 4.0

33rd International Symposium on Algorithms and Computation (ISAAC 2022).

Editors: Sang Won Bae and Heejin Park; Article No. 56; pp. 56:1–56:13

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

number of cops required to catch the robber on G . Cops and robber has been studied on several classes of graphs and many variants have been studied where cops are given different restrictions or abilities (see [6] for an overview of the area). Meyniel conjectured that in general, $c(G) = O(\sqrt{n})$.

Zombies and Survivor is a variant of the game of Cops and Robber. The deterministic version of Zombies and Survivor was first introduced by Fitzpatrick et al. [13]. In this game, each cop (thought of as a *zombie*) is restricted to move along an edge incident to its current position and belonging to a shortest path to the robber (thought of as the *survivor*). Moreover, the zombie is active, in the sense that it must move on its turn. If there is more than one shortest path between a given zombie and the survivor, the zombie can choose which path to follow. Since the zombies are active, the starting configuration of the zombies actually plays a role in determining whether a survivor is caught. For example, if several zombies are placed on the same vertex in a cycle with more than 4 vertices, then they will never catch a survivor. However, 2 zombies can be strategically placed in order to always catch a survivor on such a cycle. As such, we define two types of zombie number, one where the zombies are strategically placed and the other where an adversary determines the initial position of the zombies. The *zombie number* $z(G)$ of a graph G is then defined as the minimum number of zombies required to catch the survivor on G , and the *universal zombie number* $u(G)$ is defined as the minimum number of zombies required to catch the survivor when the starting configuration of the zombies is determined by an adversary. The above example shows that $z(G) = 2$ and $u(G) = \infty$ when G is a cycle on more than 4 vertices. Since a cop has more power than a zombie, we have $c(G) \leq z(G) \leq u(G)$. From this observation, we get that zombie-win graphs are also cop-win graphs.

In their paper, Fitzpatrick et al. [13] provide an example showing that if a graph is cop-win, then it is not necessarily zombie-win. They provide a sufficient condition for a graph to be zombie-win. They also establish several results about the zombie number of the Cartesian product of graphs.

The main aspect that makes different variants of these pursuit-evasion problems quite challenging is the fact that the cop number and zombie number is not a monotonic property with respect to subgraphs. For example, both the cop number and zombie number of a clique is 1 but the cop number and zombie number of a cycle on more than 3 vertices is 2.

1.1 Contributions

In this paper, we first consider the deterministic version of Zombies and Survivor. We then turn our attention to a deterministic variant which we call *Lazy Zombies and Survivor*. In this variant, a zombie does not need to move on its turn. The *lazy zombie number* $z_L(G)$ of a graph G is therefore defined as the minimum number of lazy zombies required to catch the survivor on G . The *universal lazy zombie number* of a graph G , denoted $u_L(G)$, denotes the minimum number of lazy zombies required to catch the survivor on G , when the starting positions of the lazy zombies are chosen by an adversary. Observe that $c(G) \leq z_L(G) \leq u_L(G)$.

We show that there exist outerplanar graphs and 2-connected outerplanar graphs G with n vertices such that $z(G)/c(G) = \Omega(n)$. This improves upon a result of Bartier et al. [4] who showed that this ratio is $z(G)/c(G) = \Omega(\log n)$ for outerplanar graphs. We also show that there exist maximum-degree-3 outerplanar graphs G such that $z(G)/c(G) = \Omega(n/\log(n))$ and there exist simple polygons whose visibility graph G is such that $z(G)/c(G) = \Omega(n)$.

Then, we show that lazy zombies are more powerful than plain zombies and less powerful than cops. Indeed, we prove that 2 lazy zombies always win (in less than $2n$ rounds) on outerplanar graphs. However, we show that there exist graphs of treewidth 2 that require 3 lazy zombies, whereas 2 cops are sufficient. We then show that k lazy zombies win after

■ **Table 1** Summary of zombie, (universal) lazy zombie, and cop numbers. The column “zombies” shows lower bounds on the zombie number. The other two columns show upper bounds. The upper bound on the cop number for h -vertex excluded minors is for *connected* excluded minors.

	zombies	(universal) lazy zombies	cops
outerplanar	$\Theta(n)$ (Thm. 1)	2 (Thm. 6)	2 ([9])
planar	$\Theta(n)$ (Thm. 1)	$O(\sqrt{n})$ (Cor. 14)	3 ([1])
genus g	$\Theta(n)$ (Thm. 1)	$O(\sqrt{gn})$ (Cor. 14)	$3 + \lfloor \frac{3g}{2} \rfloor$ ([23])
treedepth k	$\Theta(n)$ (Thm. 1)	k (Cor. 10, Thm. 11)	$(k/2) + 1$ ([16])
treewidth k	$\Theta(n)$ (Thm. 1)	$(k + 1) \log n$ (Cor. 10, Lem. 12)	$(k/2) + 1$ ([16])
h -vertex excl. minor	$\Theta(n)$ (Thm. 1)	$O(h\sqrt{hn})$ (Cor. 14)	$\frac{1}{2}(h - 1)(h - 2)$ ([3])

$O(n^{2k})$ rounds on graphs with treedepth k . Finally, we highlight a few implications stemming from this upper bound such as $(k + 1) \log n$ lazy zombies win on graphs with treewidth k , $O(\sqrt{n})$ lazy zombies are always sufficient to win on planar graphs, $O(\sqrt{gn})$ lazy zombies win on graphs with genus g and $O(h\sqrt{hn})$ lazy zombies win on all connected graphs G with any excluded h -vertex minor H . Our upper bounds on lazy zombie numbers still hold for universal lazy zombies. These results are summarized in Table 1. Further details together with missing proofs can be found in the full version of the paper [7].

2 Preliminaries and Notation

Let G be a simple and undirected graph with vertex set $V(G)$ and edge set $E(G)$. Throughout this paper n will be used for $|V(G)|$. Given a subset $S \subseteq V(G)$, we denote the graph induced by S as $G[S]$. Given two vertices u and v in $V(G)$, we denote the shortest path from u to v as $\pi_G(u, v)$ and its length¹ as $d_G(u, v)$. If there is no path from u to v , then the length of the path is infinite. The diameter of a connected graph G , denoted as $\text{diam}(G)$, is $\max\{d_G(u, v) : u, v \in V(G)\}$.

Recall that $c(G)$ is the cop number of G , $z(G)$ (resp. $u(G)$) is the zombie number of G (resp. universal zombie number) and $z_L(G)$ (resp. $u_L(G)$) is the lazy zombie number of G (resp. universal lazy zombie number). If G is disconnected, then the cop (resp. zombie, lazy zombie) number of G is simply equal to the sum of the cop (resp. zombie, lazy zombie) numbers of its connected components. Therefore, in this paper, we assume that G is connected.

In their paper, Fitzpatrick et al. [13, Figure 5] provide an example of a graph with zombie number 1, where the zombie has to start on a specific vertex to win, which is an example where $z(G) = 1$ and $u(G) = \infty$. As alluded to in the introduction, lazy zombies are more powerful than normal zombies and less so than cops. Thus, we observe that $c(G) \leq z_L(G) \leq z(G)$.

All the pursuit-evasion games we describe in this paper consist of a sequence of rounds, each of which consists of two turns. For each round $i \geq 0$, the zombies play first (zombies’ turn) and then the survivor plays (survivor’s turn). In round 0, during the zombies’ turn, the zombies choose their starting position (or an adversary assigns one to them), and then, during the survivor’s turn, the survivor chooses its starting position. Then the zombies move (or wait if the version of the game allows them to) and the survivor moves (or waits if they decide to) in the subsequent rounds.

¹ The length of the shortest path is the number of edges on the path.

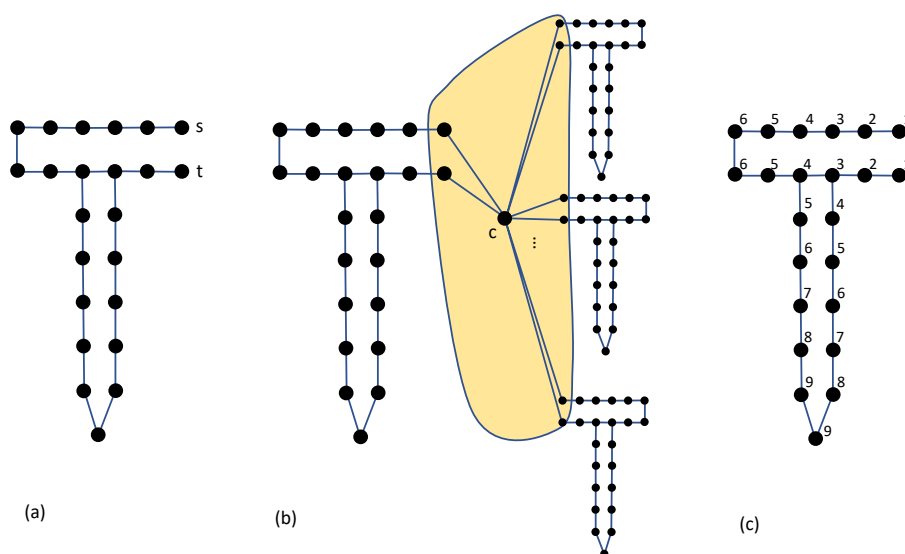
When stating results about (universal) (lazy) zombie numbers, we sometimes use asymptotic notation. In this notation, lower-bound asymptotics (Ω or Θ) will be a lower bound on the maximum-valued graph of the given class. For instance, “for outerplanar graphs G , $z(G) \in \Theta(n)$ ” means not only is the zombie number at most some constant times n for every outerplanar graph, but also the zombie number is at least some constant times n for some family of outerplanar graphs. Throughout the paper, unless specified, the base of the logarithmic function $\log(\cdot)$ is 2.

3 Linear bound on zombie number

In their paper, Fitzpatrick et al. ask how large the ratio $z(G)/c(G)$ can be [13, Question 19]; they note that they have not observed any graph with a ratio that exceeds 2. Here we show that this ratio can be infinite and of size $\Omega(n)$, and we show this even for outerplanar graphs of fixed radius. In independent work, Bartier et al. showed that this ratio can be infinite and of size $\Omega(\log n)$, for outerplanar graphs [4].

► **Theorem 1.** *Let $k \geq 2$ be an integer. Then there is a connected outerplanar graph G_k with $23k + 1$ vertices that requires at least k zombies.*

Proof. Let H be the 23-vertex graph shown in Figure 1a. H has two distinguished vertices s and t . To form the graph G_k , first take k disjoint copies H_1, H_2, \dots, H_k of H , with H_i having distinguished vertices s_i and t_i ; to this add a vertex c that is connected to each s_i and each t_i (see Figure 1b).



■ **Figure 1** Construction of an outerplanar graph requiring a linear number of zombies. (a) The component graph H . (b) Connecting components into the graph G_k . (c) Each vertex of some H_j labelled with its distance to c .

By construction, G_k has $23k + 1$ vertices. Suppose that $k - 1$ or fewer zombies play on G_k . This means that in round 0 there is some copy H_i of H that contains no zombie; the survivor chooses the vertex adjacent to s_i in H_i as its starting position, and will stay in H_i forever.

Each zombie will therefore first take a shortest path to c from wherever it starts, as this is the only way to get to H_i . Consider the time unit on which a zombie reaches c ; this can be anywhere from 0 to 9, as the zombie could start on c , and 9 is the radius of G_k (and c is the center). See Figure 1c. At one time unit later, the zombie will move to s_i or t_i , whichever is closer to the survivor. We call this time the *arrival time* of the zombie; arrival times are all between 1 and 10, inclusive.

The survivor's strategy is to walk in H_i away from s_i until it reaches a vertex of degree three. At this point they walk along the 13-cycle in H_i , starting in the direction of the vertex of degree two. They continue walking this cycle forever.

The zombies will arrive on s_i if their arrival time is at most five, as this is the closest vertex to the survivor at this time. These zombies will follow the survivor's path. If a zombie has arrival time six or more, it arrives on t_i . These zombies will pursue the survivor by first walking to the 13-cycle and then following the survivor around it.

Therefore, since $k - 1$ zombies are insufficient to capture the survivor on G_k , at least k are required and the lemma is proved. ◀

Note that a linear number of zombies always suffices for a graph, as we could use n zombies and initially place one on each vertex (or perhaps leave one free for the survivor). Thus we have shown that for general or for outerplanar graphs, $z(G) \in \Theta(n)$. Since the cop number for outerplanar graphs is at most two 2, the ratio $z(G_k)/c(G_k)$ is $k/2 = (n - 1)/46 \in \Theta(n)$.

Modifications of the construction in the proof of Theorem 1 work for other graph classes, as illustrated by the following theorems.

▶ **Theorem 2.** *Let $k \geq 2$ be an integer. There is a 2-connected outerplanar graph G_k with $30k + 1$ vertices that requires at least k zombies.*

▶ **Theorem 3.** *Let $k \geq 2$ be an integer. There is a connected graph G_k with $15k$ vertices that requires at least k zombies.*

▶ **Theorem 4.** *Let $k \geq 2$ be an integer. There is a maximum-degree-3 connected outerplanar graph G_k with at most $25k + 16k \lceil \log k \rceil - 1$ vertices that requires at least k zombies.*

Theorem 4 gives us $n \in O(k \log k)$, or $n \leq ck \log k$ for some constant c . Hence, we have $\frac{n}{\log n} \leq \frac{n}{\log k} \leq ck$, or $k \geq \frac{n}{c \log n}$. Since k zombies are required, this gives us a lower bound of $\Omega(\frac{n}{\log n})$ on the zombie number of bounded-degree graphs.

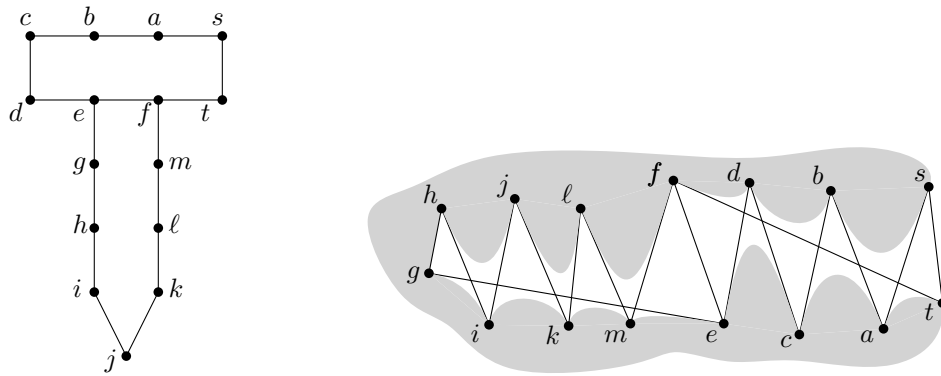
3.1 Polygon visibility graphs

A *polygonal chain* is a finite sequence V of points v_1, v_2, \dots, v_n in \mathbf{R}^2 (called *vertices*) along with the line segments $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$ (called *edges*). A polygonal chain is called *closed* if $v_1 = v_n$, and *simple* if no two edges intersect except consecutive edges intersecting at their common vertex. A closed simple polygonal chain divides the plane into a finite *interior* and infinite *exterior*. A *simple polygon*, or simply *polygon*, is a closed simple polygonal chain along with its interior.

The *visibility graph* of a simple polygon P [10, 18] is a graph G where $V(G)$ is the set of vertices of the polygon, and

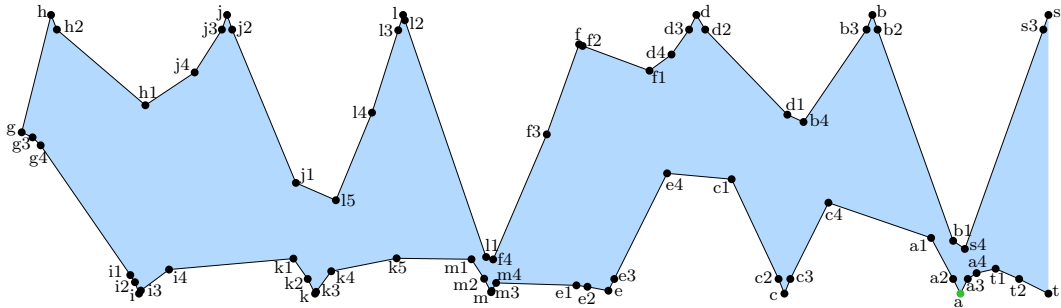
$$E(G) = \{(v_i, v_j) \mid \text{the segment } v_iv_j \text{ does not intersect the exterior of } P\}.$$

In particular, this means that every edge of the polygon is an edge of the visibility graph, but the visibility graph has other edges corresponding to segments that traverse the interior and possibly boundary of P . Visibility graphs are of interest in discrete geometry and have applications, for instance, in motion planning and shape analysis [8, 10].



(a) A graph fragment for proving (b) An embedding of the graph fragment in the visibility graph of a polygon fragment.

■ **Figure 2** Creation of a polygon fragment to prove a lower bound.



■ **Figure 3** The polygon fragment Q . Note that at this scale, the dots for the vertex-pair $(i, i3)$ merge into what can appear to be a single dot. This also happens for $(k, k3)$ and $(m, m3)$.

Here we show that there is a linear bound on the zombie number of visibility graphs. The proof is messy in that it involves a large polygonal chain with relatively precise vertex locations in order to get a visibility graph with the desired properties. However, it is inspired by the proof of Theorem 1. Consider the graph fragment in Figure 2a. One way to embed this graph inside a polygon visibility graph is sketched in Figure 2b. Complicating matters is that we cannot get the required non-edges without placing vertices inbetween those shown in Figure 2b. Once those non-visibilitys are worked out, we get the polygon fragment Q shown in Figure 3. For reproducibility, the exact vertex locations of Q are given in the full version of the paper [7].

To form a polygon whose visibility graph requires at least k zombies, we will connect k copies of Q , denoted Q_1, Q_2, \dots, Q_k , placed in a geometric configuration where the only vertices of Q_i visible to Q_j are s_i (the copy of s in Q_i) and t_i (similar). Thus S , the collection of all s_i 's and t_i 's, will form a clique in the visibility graph. This is done by taking a small sliver of a circular arc and placing the $2k$ vertices of S evenly along it. If the sliver is small enough, any vertices inside $Q_i \setminus S$ will see only the relative interior of the polygon edge $t_k s_1$. (Another method of ensuring this is to scale each Q_i up in its x -coordinate, effectively pushing interior vertices away from $s_i t_i$. Such a scaling operation does not affect the visibility graph of Q_i .)

The proof that this polygon has zombie number k now roughly follows that of Theorem 1. Suppose that less than k suffices. Then, in the zombies' initial placement, there will be one copy of Q , say Q_i , that has no zombies in it. Start with the survivor on vertex a of Q_i (refer to Figure 2a for the labelling of the vertices).

Q is constructed so that it takes at most 6 turns for a zombie to leave the fragment (or 5 turns at most to get to s or t). This means that if the survivor stays in Q_i for 6 turns (it will), all zombies will have arrived in Q_i . The survivor's strategy will be to walk from a to b , to c , to d , e , g , h , i , j , k , l , m , and f . Unlike in Theorem 1, the survivor cannot now loop back to e (they might be caught by a zombie) but must instead move to $t = t_i$. Once at t_i , the survivor chooses some s_j where $i \neq j$, and moves there. Next they can move to a_j and start the same walk in Q_j as it did in Q_i . It may continue in this way *ad infinitum*.

Since Q has 69 vertices, we have shown the following.

► **Theorem 5.** *Let $k \geq 2$ be an integer. Then there is a polygon P_k with $69k$ vertices whose visibility graph requires at least k zombies.*

A linear number of zombies will always work (e.g. $n/3$ of them starting on every third vertex), so the maximum zombie number of the visibility graph of a polygon with n vertices is $\Theta(n)$.

There is a related problem that asks for the zombie number of a *point-visibility graph* of a polygon, which is the infinite graph G where $V(G)$ is taken to be the *points* of the polygon, not simply the vertices. Edges are then defined as in the visibility graph. This problem involves more geometry than the other problems we have studied. Here it is not clear that there are polygons with a point-visibility graph zombie number higher than one.

4 The Lazy Zombie Number of Outerplanar Graphs is 2

In the previous section, we showed that $\Omega(n)$ zombies are sometimes necessary to catch a survivor on an outerplanar graph. In this section, we show that 2 lazy zombies are always sufficient to catch the survivor on outerplanar graphs. Observe that two lazy zombies are sometimes necessary to catch a survivor on an outerplanar graph since a single lazy zombie cannot win on a 4-cycle.

► **Theorem 6.** *Let G be a connected outerplanar graph. Then $z_L(G) \leq 2$ and 2 lazy zombies can catch the survivor in less than $2n$ rounds. This bound is tight in the worst case.*

Proof. We only present a proof sketch.

We first modify G by replicating cut vertices and cut edges, and adding chords, so as to make it 2-connected. We start one lazy zombie, the *stationary* lazy zombie, on the end b_j of a chord $b_i b_j$. This lazy zombie will capture the survivor if the survivor moves to b_i or b_j but otherwise will not move. We refer to the vertices of G where the survivor is known to be restricted as the *survivor territory*. The other lazy zombie (denoted z_2), the *advancing* lazy zombie, also starts at b_j .

Basically, on each turn, the advancing lazy zombie moves along the outerface towards the survivor. Suppose that at some turn, this lazy zombie is at the boundary of survivor territory at a vertex b_l which is on a chord $b_l b_k$. Assume by symmetry that b_l is counterclockwise of b_j and clockwise of b_i (See Figure 4a). Then, if the survivor is counterclockwise from b_l to b_k (in $S \setminus S'$ in the figure), we switch the roles of the lazy zombies, with a lazy zombie stationary on $b_l b_k$, and we have reduced the survivor territory. On the other hand, if the survivor is counterclockwise from b_k to b_i (in S' in the figure), the advancing lazy zombie moves to b_k , reducing the survivor territory (to S'). At no point will there be a chord from the survivor territory to the chain from b_j counterclockwise to b_l .

Since each step of the advancing lazy zombie into the survivor territory reduces the survivor territory, and no lazy zombie repeats a move to a vertex, the survivor will be captured in at most $2n$ rounds. ◀

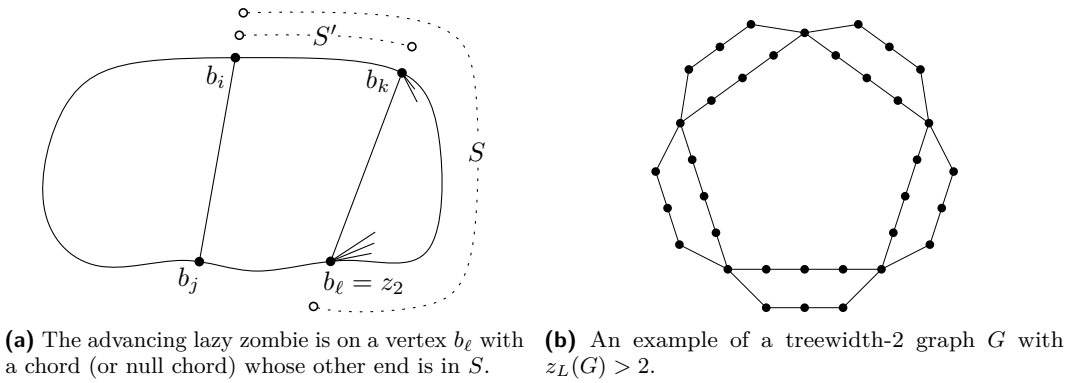


Figure 4

► **Corollary 7.** *Let G be a connected outerplanar graph. Then $u_L(G) \leq 2$. This bound is tight in the worst case.*

Outerplanar graphs are a subset of the treewidth-2 graphs, but Theorem 6 cannot be generalized to treewidth-2. Figure 4b shows a graph with treewidth two that requires three lazy zombies. This example shows a distinction between the lazy zombie number and the cop number of a graph since 2 cops are sufficient for a graph of treewidth 2 [16]. We will study general treewidth- k graphs in Section 5.

5 Cut-decomposable Graphs and Lazy Zombies

In this section, we explore the relationship between lazy zombie numbers and various graph parameters. We first define some of these graph parameters and some notation most of which appears in Diestel [11]. We then present the general approach, and finally, we outline some of the consequences of our approach.

Let T be a tree rooted at a vertex r . For a vertex $v \in V(T)$, we denote the unique path from v to r as $\pi_T(v)$. The depth of v , $d_T(v) := d_T(v, r)$, is the length of the path from v to r . If a vertex $u \in V(T)$ is in $\pi_T(v)$ then u is an ancestor of v and v is a descendant of u . Note that v is an ancestor and descendant of itself. The height of T is defined as $H_T := \max\{d_T(v) : v \in V(T)\}$. The subtree of T rooted at v is denoted as $\Lambda_T(v)$. The height of $\Lambda_T(v)$ is defined as $H_T(v) := \max\{d_T(x) - d_T(v) : x \in \Lambda_T(v)\}$. The closure of T , denoted as $clos(T)$, is $T \cup \{uv : u \text{ is an ancestor of } v \text{ in } T\}$. The *treedepth* of a connected graph G , which we denote as $td(G)$, is 1 plus the minimum height H_T over all trees T defined on $V(G)$ such that $G \subseteq clos(T)$.

A *tree decomposition* of a graph G is a pair (T, \mathcal{B}) where T is a tree and $\mathcal{B} = \{B_x \subseteq V(G) : x \in V(T)\}$, where each B_x is a subset of $V(G)$ indexed by the nodes of T . The set B_x is sometimes referred to as a *bag*. The following properties must be satisfied:

- For every $v \in V(G)$, $\{x \in V(T) : v \in B_x\}$ induces a non-empty subtree of T .
- For every $uv \in E(G)$, $\exists x \in V(T)$ such that u and v are both in B_x .

The *width* of a tree decomposition is 1 less than the cardinality of the largest bag. The *treewidth* of a graph G , which we denote as $tw(G)$, is the minimum width over all tree decompositions of G .

A *cut decomposition* of a graph G is a pair (X, \mathcal{C}) where X is a rooted tree and $\mathcal{C} = \{C_x \subseteq V(G) : x \in V(X)\}$, where each C_x is a subset of $V(G)$ indexed by the nodes of X . The following properties must be satisfied:

- For every $v \in V(G)$, there is a unique $x \in V(X)$ such that $v \in C_x$.
- For every $uv \in E(G)$, $\exists x, y \in V(X)$ such that $u \in C_x$, $v \in C_y$ and x is an ancestor of y in X .
- For each non-leaf node $y \in X$, C_y is a cut-set of $G[Y]$ where $Y = \bigcup_{x \in \Lambda_X(y)} C_x$.

We will refer to the set C_x as the *container* of x to avoid confusion with a bag of a tree decomposition. We will refer to the size of the largest container as the width of the cut decomposition tree, denoted as $cdw(X)$.

Throughout this section, we will assume that (X, \mathcal{C}) is a cut decomposition of a graph G where $|V(G)| = n$. We will refer to X as a *cut decomposition tree*. We will refer to the vertices of G as vertices and the vertices of X as nodes, in an attempt to make the distinction clear. We will use u, v to refer to vertices of G and x, y to refer to nodes in X (or nodes in a tree decomposition). For a node $y \in X$, we define the *component* of y to be $G[Y]$ where $Y = \bigcup_{x \in \Lambda_X(y)} C_x$. We slightly abuse notation and refer to the component of y as $G[\Lambda_X(y)]$. Intuitively, a cut decomposition tree is a decomposition of a graph by cuts where an internal node x of the tree X represents a cut set of the graph $G[\Lambda_X(x)]$. The container of the root of the tree contains either the entire vertex set of G , or the vertices of a cut set of G . If it contains a cut, then the children of the root recursively correspond to the different connected components of the graph that result when the cut is removed. Cut decompositions are related to both tree decompositions and treedepth (see Lemma 12).

We define the *load* of a node x in X as:

$$load(x) = \begin{cases} |C_x| & \text{if } x \text{ is a leaf,} \\ |C_x| + \max_y load(y) & \text{otherwise, where } y \text{ is a child of } x. \end{cases}$$

The load of a cut decomposition is defined as the load of the root of the cut decomposition tree. We define the load of a graph G , denoted as $load(G)$, to be the minimum load among all cut decompositions of G . We will show that $load(G)$ is a sufficient number of lazy zombies to catch a survivor in G . We define $time(x)$, where $x \in X$, to help us bound the number of rounds it takes for the lazy zombies to capture the survivor.

$$time(x) = \begin{cases} |C_x|(diam(G) - 1) + 1 & \text{if } x \text{ is a leaf,} \\ \max_y time(y)(|C_x|(diam(G) - 1) + 1) & \text{otherwise, where } y \text{ is a child of } x. \end{cases}$$

The time of the root is an upper bound on the number of rounds it takes the lazy zombies to capture the survivor.

In the following, each lazy zombie z_i may be *assigned* to a vertex v of G . The strategy of the lazy zombie will be the following. If the lazy zombie is not assigned to any vertex, then on its turn to move, it remains at its current location. A lazy zombie assigned to a vertex v has the following behavior: on its turn, it moves off its current vertex u to an adjacent vertex w only if there exists a w that is closer to both v and the survivor. This is precisely where we use the power of a lazy zombie to stand still where regular zombies cannot. Because a shortest path from z_i 's location to v has at most $diam(G)$ edges, the survivor can encounter vertex v at most $diam(G) - 1$ times (at or after the time z_i was assigned to v) without being immediately caught. Lazy zombies can and will be reassigned to different vertices during the game.

We proceed by induction. The following lemma establishes the basis.

► **Lemma 8.** *Let G be a connected graph with cut decomposition (X, \mathcal{C}) . Suppose that the survivor is restricted to the vertices of $G[C_x]$ for some leaf x in X . Then, $load(x)$ lazy zombies, starting from anywhere in G , can capture the survivor in at most $time(x)$ rounds.*

► **Theorem 9.** *Let G be a connected graph with cut decomposition (X, \mathcal{C}) . Suppose that x is a node of X and the survivor is restricted to the vertices in $G[\Lambda_X(x)]$. Then, $\text{load}(x)$ lazy zombies, starting from anywhere in G , can capture the survivor in at most $\text{time}(x)$ rounds.*

Proof. We prove the theorem by induction on $H_X(x)$, the height of $\Lambda_X(x)$. The basis for the induction, $H_X(x) = 0$, i.e. when x is a leaf, follows from Lemma 8.

We assume that $\text{load}(x)$ lazy zombies are sufficient to catch the survivor in $\text{time}(x)$ rounds when the survivor is restricted to $G[\Lambda_X(x)]$, where $H_X(x) \leq k$ for $k \geq 0$. We now proceed with the case when $H_X(x) = k + 1$. Let c be the maximum load of a child of x , and d be the maximum time for a child of x . We allocate lazy zombies z_1, z_2, \dots, z_c to the children of x . These lazy zombies are initially unassigned to any specific vertex but will be assigned to specific vertices depending on the survivor's moves. Note that it is not necessarily the case that we need to use this many lazy zombies, but this number is always sufficient. We assign lazy zombies $z_{c+1}, z_{c+2}, \dots, z_{c+|C_x|}$, each to a different vertex of C_x , respectively. Since $\text{load}(x) = |C_x| + c$, we have a sufficient number of lazy zombies.

The survivor may now encounter each vertex of C_x at most $(\text{diam}(G) - 1)$ times without immediately being caught in the next round, which again follows from the upper bound of $\text{diam}(G)$ edges on any shortest path between two vertices in G . Before the survivor's first encounter with a vertex of C_x , or between successive visits of vertices in C_x , or after the last visit, the survivor is restricted to the vertices of the component of the subtree rooted at exactly one child y of x . This follows from the fact that C_x is a cut set for $G[\Lambda_X(x)]$. We apply the inductive hypothesis on $G[\Lambda_X(y)]$, since $H_X(y) \leq k$. By the inductive hypothesis, we know that the survivor is caught after $\text{time}(y)$ rounds if the survivor remains in $G[\Lambda_X(y)]$. Therefore, the survivor must leave $G[\Lambda_X(y)]$ after $\text{time}(y) - 1 \leq d - 1$ steps, otherwise it is caught.

Each time it enters one of these subtrees, we assign the lazy zombies z_1, z_2, \dots, z_c to (specific) vertices in that subtree's component. Since c is the maximum load of any child of x , we have a sufficient number of lazy zombies. By the inductive hypothesis, this number of lazy zombies suffices to either catch the survivor if the survivor remains in the component for d steps or force the survivor out of the component of a child of x and back into C_x in at most $d - 1$ steps.

The survivor's walk length is therefore at most $(|C_x|(\text{diam}(G) - 1) + 1) + (d - 1) \times (|C_x|(\text{diam}(G) - 1) + 1)$. The first term is the number of rounds the survivor can spend in C_x until it is caught. For the second term, we note that the survivor can enter $G[\Lambda_X(y)]$ where y is a child of x at most $(|C_x|(\text{diam}(G) - 1) + 1)$ times. Each time it enters $G[\Lambda_X(y)]$ it must return to a vertex in C_x in $d - 1$ rounds, otherwise the survivor is caught on the d th round. Therefore, we have that the survivor is caught after $(|C_x|(\text{diam}(G) - 1) + 1) + (d - 1)(|C_x|(\text{diam}(G) - 1) + 1) \leq d(|C_x|(\text{diam}(G) - 1) + 1) = \text{time}(x)$. ◀

Among all cut decomposition trees realizing the load of G , we denote the value of the minimum height of such a tree by $\text{cdh}(G)$. Among all cut decomposition trees whose load is $\text{load}(G)$ and whose height is $\text{cdh}(G)$, we denote the value of the minimum width among all such trees by $\text{cdw}(G)$.

► **Corollary 10.** *Given a connected graph G , $u_L(G) \leq \text{load}(G)$ and $\text{load}(G)$ lazy zombies can catch the survivor in at most $(\text{cdw}(G)(\text{diam}(G) - 1) + 1)^{\text{cdh}(G)+1}$ rounds.*

Proof. We only present a proof sketch.

Let X be a cut decomposition tree with root r , with load $\text{load}(G)$, with height $\text{cdh}(G)$ and with width $\text{cdw}(G)$. Theorem 9 implies that $u_L(G) \leq \text{load}(G)$. Using Theorem 9 and the recursive definition of time , we show by induction that $\text{time}(r)$ is at most $(\text{cdw}(G)(\text{diam}(G) - 1) + 1)^{\text{cdh}(G)+1}$. ◀

Recall that $td(G)$ denotes the treedepth of G .

► **Theorem 11.** *For any connected graph G , $load(G) = td(G)$*

For $0 < \alpha < 1$, a cut set $S \subseteq V(G)$ of G is an α -separator if every connected component of $G[V(G) - S]$ contains at most αn vertices. The size of the α -separator is the cardinality of S . We highlight the relationship between the sizes of separators, treedepth, and treewidth.

► **Lemma 12.** *Let G be a graph. Let $s_G : \{1, \dots, n\} \rightarrow \mathbb{N}$ be a function defined as*

$$s_G(i) = \max_{A \subseteq V(G), |A| \leq i} \min\{|S| : S \text{ is a } \frac{1}{2}\text{-separator of } G[A]\}.$$

Then $s_G(n) \leq load(G) = td(G) \leq \sum_{i=0}^{\log n} s_G(n/2^i) \leq (tw(G) + 1) \log n$.

Proof. The inequality $s_G(n) \leq td(G) \leq \sum_{i=0}^{\log n} s_G(n/2^i)$ is proven in Lemma 6.6 in [19]. Since it was shown by Robertson and Seymour [22] that $s_G(i) \leq tw(G) + 1$ for all $i \in [1, n]$, we have that $\sum_{i=0}^{\log n} s_G(n/2^i) \leq (tw(G) + 1) \log n$. The equality $load(G) = td(G)$ is proven in Theorem 11. ◀

$s_G(n)$ is sometimes called the separation number of G . The bound in Lemma 12 is tight in certain cases. For example, the treewidth of a path on n vertices is 1 whereas the treedepth is $\Theta(\log n)$. However, for certain classes of graphs, we can remove the $\log n$ term on the upper bound in Lemma 12. Essentially, if $s_G(n/2^i) \leq cs_G(n)/2^i$ for some constant c , then $\sum_{i=0}^{\log n} s_G(n/2^i) \leq cs_G(n) \sum_{i=0}^{\log n} 1/2^i \leq 2cs_G(n) \leq 2c(tw(G) + 1)$. Thus, we have that $td(G)$ is $O(tw(G))$ in this case. Informally, this happens when the size of a separator for any subgraph of size i is at most i^ϵ , for $0 < \epsilon < 1$. This is summarized by the following:

► **Corollary 13** (Corollary 6.2 in [19]). *Let $0 < \alpha < 1$, let $c > 0$ be a constant and let \mathcal{G} be a hereditary class of graphs such that every $G \in \mathcal{G}$ with n vertices has $tw(G) \leq cn^\alpha$, then every $G \in \mathcal{G}$ has $td(G) \leq \frac{c}{1-2^{-\alpha}} n^\alpha$.*

Treewidth, treedepth and separators are well-studied graph parameters. We highlight a few of the implications of our bound that $u_L(G) \leq td(G)$. The interested reader should consult the following comprehensive surveys on this topic [5, 19, 15, 12]. Using Corollary 13 with separator theorems on various classes of graphs [17, 14, 2], we get the following results.

► **Corollary 14.** *For connected planar graphs G , $u_L(G)$ is $O(\sqrt{n})$. For connected genus- g graphs G , $u_L(G)$ is $O(\sqrt{gn})$. For connected graphs G with any excluded h -vertex minor H , $u_L(G)$ is $O(h\sqrt{hn})$. In all these cases, the lazy zombies can catch the survivor in at most $n^{O(\log n)}$ rounds.*

Although $load(G)$ is an upper bound on $u_L(G)$, it is by no means tight. If G is a clique, then $load(G) = n$, but only 1 lazy zombie suffices to catch a survivor. We try to leverage this idea to get a slightly tighter bound.

By assigning one lazy zombie to each clique in a container of the cut decomposition rather than one lazy zombie to each vertex in the container, we can improve the upper bound. This idea leads to an alternative definition of load for a cut decomposition which we call $load^*$. In this definition, for $S \subseteq V(G)$, $\theta(S)$ is the clique cover number of the induced graph of G on the vertices of S .

$$load^*(v) = \begin{cases} \theta(C_v) & \text{if } v \text{ is a leaf,} \\ \theta(C_v) + \max_w load^*(w) & \text{otherwise, where } w \text{ is a child of } v. \end{cases}$$

If C_v is an independent set then $\theta(C_v) = |C_v|$. Thus, without knowing more about the cuts, load^* is no more useful than load . However, load^* is a substantial improvement for some graphs. For example, if G is a clique then $\text{load}^*(G) = 1$ which is optimal. The corresponding notion of time is

$$\text{time}^*(v) = \begin{cases} \theta(C_v)\text{diam}(G) + 1 & \text{if } v \text{ is a leaf,} \\ (\theta(C_v)\text{diam}(G) + 1) \max_w \text{time}^*(w) & \text{otherwise, where } w \text{ is a child of } v. \end{cases}$$

► **Theorem 15.** *Let G be a connected graph with cut decomposition (X, \mathcal{C}) . Suppose that v is a vertex of X and the survivor is restricted to the component of v . Then, $\text{load}^*(v)$ lazy zombies can capture the survivor in at most $\text{time}^*(v)$ rounds.*

► **Corollary 16.** *Given a connected graph G , $u_L(G) \leq \text{load}^*(G) \leq \text{load}(G)$ and $\text{load}^*(G)$ lazy zombies can catch the survivor in at most $(\theta(G)(\text{diam}(G) + 1))^{\text{cdh}(G)+1}$ rounds.*

References

- 1 M. Aigner and M. Fromme. A game of cops and robbers. *Discret. Appl. Math.*, 8(1):1–12, 1984.
- 2 N. Alon, P. Seymour, and R. Thomas. A separator theorem for graphs with an excluded minor and its applications. In H. Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 293–299. ACM, 1990.
- 3 T. Andreae. On a pursuit game played on graphs for which a minor is excluded. *J. Comb. Theory, Ser. B*, 41(1):37–47, 1986.
- 4 V. Bartier, L. Bénéteau, M. Bonamy, H. La, and J. Narboni. A note on deterministic zombies. *Discret. Appl. Math.*, 301:65–68, 2021.
- 5 H. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998.
- 6 A. Bonato and R. Nowakowski. *The Game of Cops and Robbers on Graphs*. AMS, 2011.
- 7 P. Bose, J.-L. De Carufel, and T. Shermer. Pursuit-evasion in graphs: Zombies, lazy zombies and a survivor. *CoRR*, abs/2204.11926, 2022. [arXiv:2204.11926](https://arxiv.org/abs/2204.11926).
- 8 H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun. *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press, May 2005.
- 9 N. Clarke. *Constrained cops and robber*. PhD thesis, Dalhousie University, Halifax, Canada, 2002.
- 10 M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008.
- 11 R. Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 12 Z. Dvořák and S. Norin. Treewidth of graphs with balanced separations. *J. Comb. Theory, Ser. B*, 137:137–144, 2019.
- 13 S. Fitzpatrick, J. Howell, M.-E. Messinger, and D. Pike. A deterministic version of the game of zombies and survivors on graphs. *Discret. Appl. Math.*, 213:1–12, 2016.
- 14 J. Gilbert, J. Hutchinson, and R. Tarjan. A separator theorem for graphs of bounded genus. *J. Algorithms*, 5(3):391–407, 1984.
- 15 D. Harvey and D. Wood. Parameters tied to treewidth. *J. Graph Theory*, 84(4):364–385, 2017.
- 16 G. Joret, M. Kaminski, and D. Theis. The cops and robber game on graphs with forbidden (induced) subgraphs. *Contributions Discret. Math.*, 5(2), 2010.
- 17 R. Lipton and R. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

- 18 T. Lozano-Pérez and M. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, 1979.
- 19 J. Nešetřil and P. Ossona de Mendez. *Sparsity – Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.
- 20 R. Nowakowski and P. Winkler. Vertex-to-vertex pursuit in a graph. *Discret. Math.*, 43(2-3):235–239, 1983.
- 21 A. Quilliot. *Jeux et pointes fixes sur les graphes*. PhD thesis, Université de Paris VI, Paris, France, 1978.
- 22 N. Robertson and P. Seymour. Graph minors. II. algorithmic aspects of tree-width. *J. Algorithms*, 7(3):309–322, 1986.
- 23 B. Schröder. The copnumber of a graph is bounded by $\lceil 3/2 \text{ genus}(g) \rceil + 3$. In J. Kosłowski and A. Melton, editors, *Categorical Perspectives*, pages 243–263, Boston, MA, 2001. Birkhäuser Boston.