



Games on Graphs: Cop and Robber, Hungry Spiders, and Broadcast Domination

Joshua Carlson, Pamela E. Harris, Peter Hollander, and Erik Insko

Abstract

There are many types of games one can play on a graph. In this chapter, we introduce three games on graphs: cops and robbers, hungry spiders, and broadcast domination. Each of these games allows for a multitude of research opportunities and further generalizations of the game, and the games can be studied and analyzed from a variety of perspectives, providing for students an immediate area of research with very little need for much technical or sophisticated mathematical background. We end the chapter by describing some courses beyond calculus and additional techniques they provide, which will allow for further study of these games from new mathematical perspectives.

Suggested Prerequisites *Curiosity and a love for games. A very basic background on graphs.*

J. Carlson (✉)

Drake University, Des Moines, IA, USA

e-mail: joshua.carlson@drake.edu

P. E. Harris

University of Wisconsin Milwaukee, Milwaukee, WI, USA

e-mail: peharris@uwm.edu

P. Hollander

Williams College, Williamstown, MA, USA

e-mail: pjh1@williams.edu

E. Insko

Florida Gulf Coast University, Fort Myers, FL, USA

e-mail: einsko@fgcu.edu

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022

E. E. Goldwyn et al. (eds.), *Mathematics Research for the Beginning Student,*

Volume 1, Foundations for Undergraduate Research in Mathematics,

https://doi.org/10.1007/978-3-031-08560-4_1

1 Introduction

Since ancient times, humans have been fascinated with games. In fact, games form an integral part of every human culture, and they provide us with a way to both exercise our intellect and socialize with others. It is no surprise, then, that mathematicians have studied ways to identify winning strategies, measure game complexity, or determine the unsolvability of games and puzzles. Luckily for us, there is absolutely no shortage of games in which such analyses can be performed. Also, as most games are often simple enough to introduce, yet challenging enough to analyze and/or solve mathematically, this setup allows a wide breadth of people with varying mathematical backgrounds to enjoy both the play and the analysis.

In this chapter, we consider games of two types: two-player and single-player games. In two-player games, two players alternate turns playing the game, and they each want to figure out a way to win the game. Classical examples of two-player games include tic-tac-toe, checkers, and chess. As expected, a single-player game is one in which a single player is given some information or resources, and they play a game to see if they can win by solving a task. The card game *solitaire* is an example of a single-player game. Moreover, games can be played using cards, tokens, or boards with pieces.

In our work, we focus on games we can play on a mathematical structure known as a graph, which is made up of vertices (dots) and edges (line segments) connecting the vertices. In order to make our approach concise, we delay all of the technical definitions and background related to graphs until Sect. 6, and any further definitions needed for a particular game are presented in their corresponding section. The three games we present include a two-player game, a single-player game, and a game that can be either.

Our first game is known as the *cop and robber game*. In this game, we place a “cop” and a “robber” on the vertices of a graph. The game begins with the cop moving from their starting vertex to an adjacent vertex (one connected via an edge), which is followed by the robber moving to an adjacent vertex from where they began. The goal of the game is to determine whether the cop can catch the robber, which is when the game ends. The second game we present is called *hungry spiders*, where one or more flies are stuck at fixed vertices in a web (a finite graph), and one or more spiders try to capture them.¹ Each round, the player who is trying to save the flies, gets to choose one edge of the graph to delete in each round, and the game ends when the player deletes enough edges to separate the spiders from the flies or when the spiders capture a fly. The last game we present is a single-player game called *broadcast domination* in which a single-player allocates “resources” on certain vertices (called “towers”). These “resources” are then propagated to neighboring vertices, and the ultimate goal of the game is to find an efficient placement of as few “towers” as possible while spreading sufficient resources to every vertex on the graph.

¹ Our hungry spiders game is based on the Greedy Spiders mobile app game.

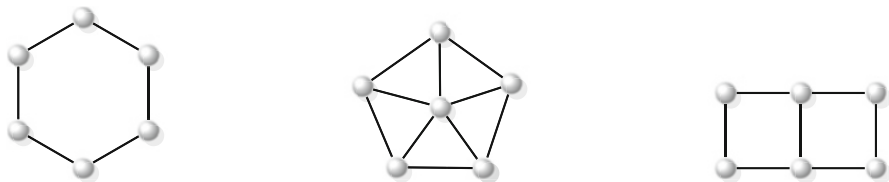


Fig. 1 Cycle graph C_6 , wheel graph W_6 , and grid graph $G_{2,3}$

Now that we know some initial information about our games, let our game playing begin!

Game 1: Cops and Robbers Select a graph from Fig. 1, draw it large on a piece of paper and grab a friend. Find two different tokens (coins or beads work great for this).

Now, here are the rules: Player 1 places their token down on a vertex, and then Player 2 places their token down on some other vertex. The players then alternate moving their tokens along the edges to a vertex next to the vertex they are currently on. Player 1 wins the game if there is ever a time when the tokens are on the same vertex, and Player 2 wins the game if this can always be avoided. Who wins the game? What if you try a different graph?

Game 2: Hungry Spiders Select a graph from Fig. 1, draw it large on a piece of paper, grab a friend, and two or three different tokens. Now, here are the rules: Player 1 (the spider) places a token down on a vertex, and then Player 2 (the fly) places a token down on some other vertex. The players then alternate turns, starting with the fly. In each round of the game, the fly gets to remove one edge from the graph, and the spider gets to move from the vertex they currently occupy along an edge to an adjacent vertex. The spider wins the game if there is ever a time when the tokens are on the same vertex, and the fly wins if there are no longer any paths that the spider can follow to reach the fly. Who wins the game? What if you try a different graph? Does the outcome depend on where the players start? What if Player 2 has to place two flies on the graph at the start? What if Player 1 gets to place two spiders? As you can see, there are many possible generalizations to make the game interesting, and we believe that studying these variations could provide a wealth of undergraduate research opportunities.

We remark that we provide Sect. 6 with a short primer on graph theory background. The reader can, as needed, refer to that section for more technical background, and once the reader begins actively researching a topic, we recommend reading the articles cited in this chapter (and conducting a literature review of the articles they cite as well) to see examples of standard proof techniques in analyzing games on graphs. We are now ready to formally introduce our games on graphs

2 The Game of Cops and Robbers

The original game of cops and robbers on graphs was introduced independently by Quilliot [9] and Nowakowski and Winkler [8]. The game of cops and robbers on a simple connected graph G proceeds as follows. Begin the game at time 0 by placing a cop and a robber on distinct vertices of G . At time 1, the robber moves from its occupied vertex to a new vertex by walking across an edge of the graph G . At time 2, the cop moves in a similar fashion. As the cop and robber alternate turns moving from vertex to adjacent vertex in $V(G)$, the cop is trying to capture the robber, and the robber is trying to elude the cop. The game comes to an end when the cop and the robber occupy the same vertex, in which case the robber is caught. An example of a sequence of moves (or turns) for a game of cops and robbers is illustrated in Fig. 2.

Challenge Problem 1 Consider the game of cops and robbers on the path graph with n vertices. If the robber and cop are placed on this graph, and the smallest number of vertices between them is m (i.e., the distance between their respective positions is $m + 1$), then what is the maximum time the game will last when the cop is playing optimally (always trying to reduce the distance between themselves and the robber)?

The previous challenge problem may lead us to incorrectly assume that the cop can always catch the robber. However, even when playing optimally this might not be the case! Consider playing the game on the cycle graph with $n \geq 4$ vertices. In this case, so long as the starting distance between the cop and the robber is greater than one, the cop will chase the robber around the graph forever and the game will never end. We call such a graph a “robber win” graph.

Research Project 1 Consider graphs on at most 10 vertices (a list of many interesting graphs is found here: <https://www.graphclasses.org/smallgraphs.html>), and determine which are robber win graphs.

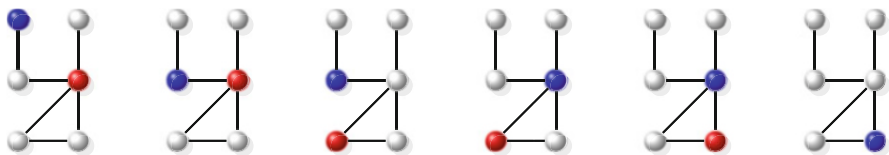


Fig. 2 A game of cops and robbers that ends at time 5 (cop is blue and robber is red)

Challenge Problem 2 Consider a variation of the game of cops and robbers in which the robber can choose to move or to remain on the same vertex each turn, while the cop always moves on their turn trying to catch the robber. Using this version of the game, consider again Challenge Problem 1, and determine the maximum time the game will last under this new constraint.

In the remainder of the section, we consider a change to the way in which the cop and robber can move as well as the game's objective. In this variant, the initial placements of the cop and robber occur as follows. First, the cop chooses an initial vertex to occupy. Then, the robber can observe the initial placement of the cop and use this information as they choose their initial vertex. For movement, suppose the cop and the robber both are able to either remain on the same vertex or freely move to any neighboring vertex on their respective turns. As for the objective, first, consider the following notion of damage introduced by Cox and Sanaei in 2019 [3].

A vertex v is said to be *damaged* whenever the robber is able to make a move from v (either staying on v or moving to a neighbor of v) without being caught during the cop's next turn. Once vertices are damaged, they cannot be undamaged. Also note that from the robber's perspective, simply moving to a vertex v is not sufficient for damaging that vertex. The robber must be able to take a turn while starting on v without immediately getting caught by the cop. For example, suppose the cop and robber start at opposite ends of the path P_{10} as in Fig. 3. If the players always move toward each other on each turn, only four vertices are damaged since the robber is caught before they can make a move from v_6 .

The idea of the *damage variant of the cops and robbers game* is that, aside from capturing the robber, minimizing damage might be the next best thing that the cop can do. In this variant, optimal play for the cop minimizes the number of damaged vertices, and optimal play for the robber maximizes that number. Note that the initial placement step plays a role in considering optimal play since the wrong initial placement could influence the number of damaged vertices. In [3], the authors define the *damage number* of a graph G , denoted $\text{dmg}(G)$, as the minimum number of vertices damaged in G over all games played where the robber plays optimally. In other words, $\text{dmg}(G)$ is the exact number of vertices damaged in G when both the cop and the robber play optimally. As a consequence, determining the damage number does not always require capture of the robber.

Observe that for a given graph G and integer k , we can prove that $\text{dmg}(G) = k$ in two steps. First, we can prove $\text{dmg}(G) \leq k$ by producing a strategy for the cop that guarantees that no more than k vertices can be damaged regardless of the robber's strategy. Second, we can prove $\text{dmg}(G) \geq k$ by producing a strategy for the robber that guarantees that at least k vertices are damaged regardless of the cop's strategy.

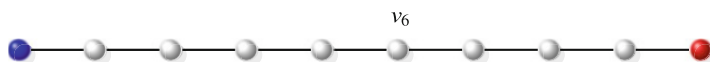
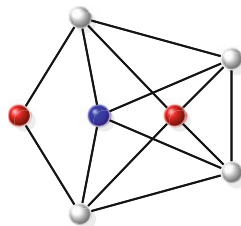


Fig. 3 The path P_{10} with cop (blue) and robber (red) starting at opposite endpoints

Fig. 4 A graph G with $\text{dmg}(G) = 1$



Challenge Problem 3 For each integer $n \geq 1$, determine with proof $\text{dmg}(P_n)$.

Challenge Problem 4 For each integer $n \geq 4$, describe a strategy for the cop that proves $\text{dmg}(C_n) \leq \lfloor \frac{n-1}{2} \rfloor$.

An interesting property of the damage variant of the cop and robber game is that even though the cop may be able to catch the robber, this may not be optimal for minimizing damage. For example, in the graph in Fig. 4, there is a strategy for the cop that guarantees capture of the robber in which multiple vertices are damaged (try to verify this). However, note that if the cop is initially placed on the blue vertex, they can catch the robber whenever the robber moves to a white vertex. Therefore, if the robber starts on a white vertex, no damage is done. Thus, the robber can choose to start on a red vertex and remain there to achieve maximum damage. Either way, the cop minimizes damage by staying in place rather than capturing the robber.

Note that by choosing a vertex of maximum degree and remaining on that vertex, the cop can always protect $\Delta(G) + 1$ vertices. This implies that for any graph G on n vertices, $\text{dmg}(G) \leq |V(G)| - \Delta(G) - 1$ [3]. We leave it to the reader to verify the following:

Challenge Problem 5 Graphs on n vertices with $\Delta(G) = n - 1$ or $\Delta(G) = n - 2$ satisfy $\text{dmg}(G) = n - \Delta(G) - 1$.

In [2], the authors show that these are not the only graphs with damage number equal to $|V(G)| - \Delta(G) - 1$. This motivates the following research project (originally posed in [3]).

Research Project 2 Give a complete characterization of all graphs G on n vertices that satisfy $\text{dmg}(G) = n - \Delta(G) - 1$.

We can also play the damage variant with k cops versus a single robber. In this version, the objectives are the same, and the two teams (cop team and robber team) alternate taking turns with all k cops moving simultaneously on their turn. The k -damage number of a graph G , $\text{dmg}_k(G)$, is defined similarly to $\text{dmg}(G)$ except k

cops play against one robber. In [2], Carlson, Egleton, Geneson, Petrucci, Reinhart, and Sen define the k -damage number, but they do not determine this number exactly for many families of graphs. This leads to the following project.

Research Project 3 Suppose we are given an arbitrary positive integer k . For which graphs G , can we determine an exact formula for $\text{dmg}_k(G)$ in terms of k and $|V(G)|$?

3 Hungry Spiders

The next set of projects concerns questions motivated by the Greedy Spiders game created in 2011 by the indie game studio Blyts. Since we will be considering variations, we call it the hungry spiders game. In this game, the graph G represents a spider web where a set of flies are stuck at a subset of vertices $F \subseteq V(G)$. The spider(s) are allowed to move about the web following a rule or strategy, traversing one edge at a time during their turn. In Blyts' Greedy Spiders game, the spiders are controlled by the computer, and each spider moves toward a nearest fly in each round of the game. The player gets to remove one edge per turn in an effort to separate the flies from the spider(s). The spider wins if it manages to reach any one of the flies, and the flies win if the spider can no longer reach any of them.

Figure 5 shows a possible web and a sequence of moves in which the fly wins. An interested reader may also watch the game in action in the official trailer using the following link: <https://youtu.be/qv0X-m1pbCw>.

We now describe the game more formally in the language of graph theory. At the start of the game, the player is presented with a simple graph $G_0 = (V, E_0)$. The spiders are placed at some subset of vertices $S_0 \subseteq V$, and the flies are stuck at a subset $F \subseteq V$ of vertices for the duration of the game. In the original version of the game, the spiders' moves follow a greedy rule, meaning they always move to a vertex that decreases the distance between them and at least one of the nearest flies. (As an aside, this particular greedy rule may not always be the optimal strategy for the spiders to employ, and later in this section we ask what other strategies the spiders might implement in different starting configurations.)

An instance of the game proceeds in rounds, and at the beginning of round i , we call the current graph $G_i = (V, E_i)$ and the current location of the spiders S_i . Each round consists of the following two steps:



Fig. 5 A sequence of moves in which the fly wins (fly in blue and spider in red)

1. The player picks an edge $e \in E_i$ to delete from $G_i = (V, E_i)$, forming the new graph $G_{i+1} = (V, E_{i+1})$
2. The spiders jointly compute their next move according to the greedy rule, and each spider moves to an adjacent vertex. Hence, their set of locations S_i becomes S_{i+1} .

Throughout the game, the player’s goal is to find a sequence of edge deletions which keep S_t from F for all $t \geq 0$. In other words, the player wants to find a sequence of edge deletions that disconnects the part of the graph containing the spiders from the part of the graph containing the flies. If any spider ever reaches a fly, the game on that graph is over, and the flies have to try again. Figure 6 shows one such case.

While creating new levels, the game developers want to categorize the difficulty level of various graphs and starting configurations, beginning with the ones that are too easy. For instance, Blyts’ game starts with two warm-up levels containing a single fly trapped on a web with a single spider.

Challenge Problem 6 Show that a single fly can win against a single spider on any graph regardless of where the spider is placed.

We note that slight variations in a graph or the placement of the spider and flies on the same graph can make the game much more challenging or change the game’s outcome.

Challenge Problem 7 Determine which of the graphs in Fig. 7 are fly-win or spider-win if both the spider and flies employ their optimal strategy.

After playing a few rounds, the reader will surely notice that if the spider can reach a vertex that is adjacent to two or more flies, then the spider will win.

Challenge Problem 8 Construct a family of graphs and configurations of flies on which the flies win when the spider employs a greedy rule that simply moves toward

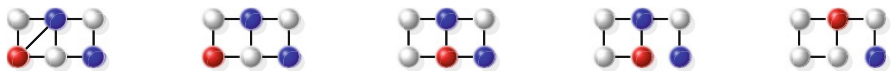


Fig. 6 A sequence of moves on a graph in which spider wins (flies in blue and spider in red)

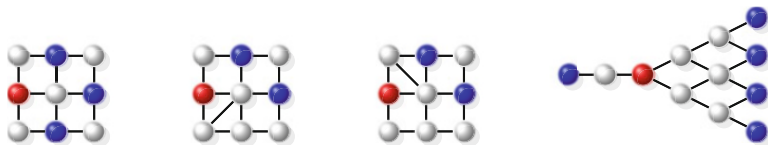


Fig. 7 Classify each game board as fly-win or spider-win given these starting configurations

a nearest fly, but not if the spider employs a different strategy that moves toward the nearest vertex that is adjacent to two or more flies. (One such example appears in Fig. 7.)

Research Project 4 Is there a succinct classification of all graphs and configurations of flies on which the flies win when the spider employs a greedy rule that simply moves toward a nearest fly, but not if the spider employs a different strategy that moves toward the nearest vertex that is adjacent to two or more flies?

Finding examples of graphs with a certain property is often easy, but classifying all graphs with that property usually takes a considerable amount of work.

Research Project 5 Classify families of fly configurations on graphs where a set of flies win if they play optimally against a single spider.

We have already noted that a graph with only one fly is always a fly-win graph. One sufficient condition in your classification might be configurations in which the spider cannot reach any vertex of the graph that is adjacent to two or more flies. Another sufficient condition might be configurations in which the flies can cut all of the edges between their component and the spider's before the spider can reach that cut.

Research Project 6 Classify families of player configurations on graphs where a set of flies win if they play optimally against two spiders.

For later levels, the game developers want graphs where the flies win, but only if they make a series of carefully chosen edge deletions.

Challenge Problem 9 Construct a family of graphs on which the flies can win but not without deleting at least four edges. Construct a family of graphs on which the fly can win but not without deleting at least n edges.

Additionally, these types of questions can be explored regarding an even broader set of graphs.

Research Project 7 Tackle the previously stated challenge problems and research projects on *multi-graphs*, i.e., graphs with multiple parallel edges connecting vertices.

A search for greedy or hungry spiders on the <https://arxiv.org>² turned up no results, and as far as we know, there are no mathematical papers studying the hungry spiders game. However, hungry spiders fits a larger category of graph problems called edge deletion problems, which ask to transform an input graph into a graph in a given, specific class of graphs by deleting a minimum number of edges. There is a wealth of literature on edge deletion problems, and to start studying their complexity, we recommend reading Yannakakis' article from 1981 and any of the many articles citing it on the SIAM website [10].

We remark that there are countless variations or additions one could make to the game. In Blyts' game, the player gets additional move types as the levels get more challenging. For instance, the player can place a decoy fly that may lure the spiders away from the real flies for a time period, or the player can delete a vertex and all its incident edges from the graph. For further study, consider the following list of possibilities or brainstorm your own:

- Change the spiders' algorithm from a greedy rule or play against another person.
- Allow the player to delete one or more vertices as well as deleting an edge in each turn.
- Place decoy flies to lure the spiders away from the real ones.
- Use a digraph as the web in which spiders must move in the direction of the arcs.
- Allow the flies the special option to teleport to a different vertex one time per round.
- Rather than alternating turns, a coin-flip or a spinner could decide who gets to go next. In this case, winning the game on a particular graph would also depend on chance.

4 (t, r) Broadcast Domination

A known generalization of the process of graph domination is known as (t, r) broadcast domination, and it is defined using the following analogy. Imagine some vertices of a grid graph represent houses, some represent cell phone towers, and we are allowed to freely choose which of the vertices are houses and which are the towers. The towers all emit some signal of a known strength. The goal of this game

² This is a repository of articles used by mathematicians to share their most recent results in a broad range of mathematical areas.

is to ensure that we can place towers so that all houses receive a minimum amount of necessary signal using as few towers as possible.

In particular, we can say that each tower gives itself signal strength t (where t is a positive integer), and its signal strength decays linearly outward from vertex to vertex as it traverses the edges of the graph. More specifically, if a tower is located on vertex u , then each vertex $v \in V(G)$ receives signal strength $t - \text{dist}(u, v)$ from that tower. If there are multiple towers whose signal reaches a single house, we add those signals together to determine the amount of signal the house receives. Given a graph G and positive integers t and r , a (t, r) broadcast dominating set of G is a set of vertices on which towers (with signal t) can be placed so that every vertex in $V(G)$ receives a total signal of at least r . The (t, r) broadcast domination number of G , denoted $\gamma_{t,r}(G)$, is the minimum size of a (t, r) broadcast dominating set in G . Intuitively, this number is the fewest number of towers needed in G to ensure that every vertex receives sufficient reception.

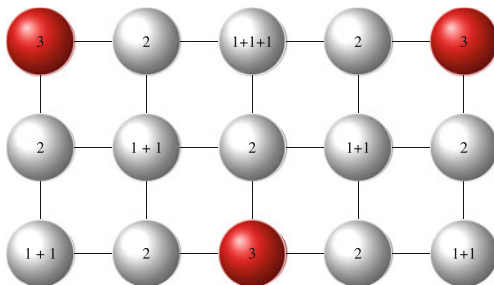
It is worth pointing out that $(2, 1)$ broadcast domination is equivalent to the standard notion of graph domination where each tower’s signal only reaches that tower’s direct neighbor, and we only require that each vertex on the graph either is a tower or is a house receiving signal from at least one tower. We refer the readers interested in a comprehensive study in (classical) graph domination to consult the text by Haynes, Hedetniemi, and Slater [7].

Figure 8 illustrates an example of a $(3, 2)$ broadcast dominating set of a 3×5 grid graph. In this example, note that the vertices of distance 2 from any tower receive low signals (of strength 1) from each tower. However, the low signals from each tower add together to give these vertices sufficient reception.

The study of (t, r) broadcast domination number of grid graphs originated with the work of Blessing, Insko, Johnson, and Mauretour in 2015 [1]. In this paper, the authors determined the broadcast domination number for various small values of t and r for $2 \times n, 3 \times n, 4 \times n$, and $5 \times n$ grid graphs. Consider the following challenge problems for an opportunity to practice broadcast domination on grid graphs.

Challenge Problem 10 Show that the vertices of the 3×5 grid graph cannot all receive minimum signal $r = 2$ from only two towers with strength $t = 3$.

Fig. 8 The 3×5 grid graph with $t = 3$ towers (shown in red) placed at positions $(1, 1)$, $(3, 3)$, and $(1, 5)$. The signal receptions are shown on the vertices



Challenge Problem 11 Describe how to find a minimum $(2, 2)$ dominating set on a $3 \times n$ grid graph for each $n \geq 3$.

Challenge Problem 12 Describe how to find a minimum $(2, 2)$ dominating set on a $4 \times n$ grid graph for each $n \geq 3$. [*Hint: it is known that the $(2, 2)$ domination number of $G_{4,n}$ is $2n - \lceil \frac{n-6}{4} \rceil$.]*

While exact formulas for some (t, r) broadcast domination numbers exist, extending this work is still an open area of research, leading to the following research project (originally posed in 2020 in [6]).

Research Project 8 Find formulas for the (t, r) broadcast domination number for grid graphs $G_{m,n}$ with $m \in \{2, 3, 4, 5\}$, $n \in \mathbb{N}$, and (t, r) different from $(2, 2)$, $(3, 1)$, $(3, 2)$, $(3, 3)$.

We can even think of converting these grid graphs into cylinders by “gluing” the right and left ends of the grids as follows: for every row, simply connect the rightmost and leftmost vertices of that row by an edge. Now, we can consider the following research project.

Research Project 9 Given a grid graph $G_{m,n}$ and values t, r , connect the vertices in the rightmost column of $G_{m,n}$ to those in the leftmost column with edges, and find formulas for the (t, r) broadcast domination number of the resulting graph.

Another potentially fruitful area of research is to find patterns for efficient broadcast domination in graph families which are altogether different from grid graphs. Consider the following challenge problem to practice thinking about broadcast domination for other graph families.

Challenge Problem 13 Let $t > r \geq 1$. What is the (t, r) broadcast domination number of a complete graph on n vertices?

There are many more families for which nothing is known about broadcast domination. The following research project provides an opportunity for great discovery.

Research Project 10 Determine the (t, r) broadcast domination number of various families of graphs. As a starting point, consider the Petersen graph, cycles, trees, and complete graphs (see Sect. 6 for definitions of these graph families).

It is also possible to study broadcast domination in the context of various common graph operations. The following challenge problem highlights how the (t, r) broadcast domination number may change after performing these operations.

Challenge Problem 14 Suppose t and r are integers with $1 \leq r < t$. Construct a graph G for which deletion of any edge results in an increase of the (t, r) broadcast domination number. Construct a graph G on which deletion of any edge does not affect the broadcast domination number.

Further study of the intricacies of vertex and edge deletion (and other graph operations) may lead to interesting findings and a greater understanding of the changes in (t, r) broadcast domination number as one transitions from one graph to another using those operations. This leads to the following research project.

Research Project 11 Study the effects of various graph operations on (t, r) broadcast domination number of a graph. How does this number change when an edge or a vertex is deleted? How does the (t, r) broadcast domination number of a graph G compare to that of its complement \bar{G} ? How about when an edge is subdivided (i.e., we add a vertex x to $V(G)$ and replace an edge (u, v) with edges (u, x) , (x, v))?

A process that has just been proposed within the past year is that of *directed broadcast domination* [5]. Given a graph G , orient each of its edges to create an orientation \vec{G} of G . Now, the process of directed broadcast domination works in \vec{G} much like in G , except we only allow signal to travel in one direction along the arcs of \vec{G} .

Note that the directed (t, r) broadcast domination number of \vec{G} may change depending on which orientation \vec{G} we choose. To this end, define the (t, r) *directed broadcast domination interval* of a graph G , denoted (t, r) DBDI(G), to be the smallest closed interval that contains all possible (t, r) directed broadcast domination numbers of \vec{G} taken over all orientations of G .

Challenge Problem 15 What is the $(3, 2)$ directed broadcast domination interval of the cycle on six vertices?

The following project in directed broadcast domination explores the bounds of this interval for various families of graphs.

Research Project 12 Given values of t and r , what are the maximum and minimum values in the (t, r) DBDI of various families of graphs?

Graphs may also be studied within a probabilistic context using randomness. The following problem highlights (t, r) broadcast domination as a problem in probability.

Challenge Problem 16 Let $t = r = 2$. For each $x \in \{1, 2, 3, 4, 5\}$, what is the probability that a randomly selected set of vertices of size x is a (t, r) broadcast dominating set of a path on 5 vertices?

One can imagine a real-life application of random (t, r) broadcast domination. For example, in some computer networks, the network “hosts” may be shuffled around randomly within a group from time to time with some arrangements more optimal than others. This creates a very applicable area of research.

Research Project 13 Given an arbitrary graph G and fixed values t and r , what is the probability that a randomly selected set of vertices is a (t, r) broadcast dominating set?

Lastly, there are many ways in which broadcast domination can be altered to produce entirely new fields of research. One potential direction for study is to alter the way in which signal decays over the edges of a graph. For example, consider the following research project.

Research Project 14 Study (t, r) broadcast domination when signal decays nonlinearly over edges. One example of this is to reduce the signal by half across every edge. Note that when $r = 1$, this process resembles *exponential domination*, described in detail in [4].

5 Further Investigation

As we have seen, many problems in analyzing games on graphs, such as the ones we presented in this chapter, can be pondered with relatively little mathematics background. This makes games on graphs an incredibly accessible area of study for students who are curious and willing to learn and play. Yet we have only presented very few of the many ways in which these games can be studied. In fact, with more background in probability, combinatorics, linear algebra, and abstract algebra, students could continue exploring the games we presented with new techniques. Such investigations could easily lead to research projects for undergraduate theses or even lead to Ph.D. dissertation projects in mathematics. We encourage students to continue in their mathematical journey, keep asking questions, but more importantly have fun while playing games!

6 A Short Primer on Graph Theory

We begin by introducing some needed mathematical background and notation. A *graph* G is defined by a set of *vertices* (dots), denoted by $V(G)$, and a set of *edges*, denoted by $E(G)$, where an edge is drawn as a line segment connecting two vertices. If the graph G is clear from context, we write V and E for the vertex and edge sets, respectively. If an edge e connects vertex u to vertex v , we say that u and v are *endpoints* of e , and we often write $e = uv$. Furthermore, every edge is said to be *incident* to its endpoints. When we say a graph is *finite*, we mean that the number of vertices is finite, and when we say the graph is *simple*, we mean that every edge must have two distinct endpoints (no loops), and there can be at most one edge incident to any pair of distinct vertices (no parallel edges). Throughout this chapter, we will always assume that G is a finite simple graph. Table 1 illustrates some common families of finite, simple graphs.

Note that the $m \times n$ *grid graph* $G_{m,n}$ has vertices arranged into an array consisting of m rows and n columns. Therefore, the grid graph as drawn in Table 1 is denoted as $G_{2,3}$. Another simple graph that is particularly important in graph theory is called the *Petersen graph*. The Petersen graph has 10 vertices and 15 edges and is illustrated in Fig. 9.

If a graph G is not a member of a well-defined family, or if it does not already have a special name, we can still describe G using a variety of terms and concepts from graph theory. For instance, we say that a graph is *connected* if, given any two vertices u and v , we can find a way to “walk” from u to v by traveling along the edges of the graph to get from one vertex to another. Note that the graphs shown in Table 1 and Fig. 9 are indeed connected. We say a graph contains a *cycle of length n* if we can find a vertex u and a “walk” along n distinct edges of the graph that allows us to begin and end at the vertex u . Note that the *wheel graph* W_6 illustrated in Table 1 contains many cycles of length three. Graphs that are connected and contain no cycles are called *trees*. The *path graph* P_n and the *star graph* S_n are the only graphs listed in Table 1 that are trees for each integer $n \geq 1$. Do note that the grid

Table 1 Each row describes a family of finite, simple graphs and depicts an example on 6 vertices.

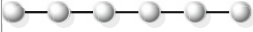
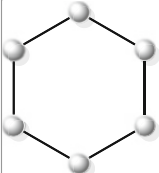
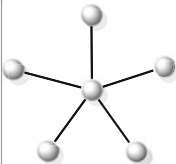
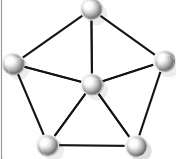
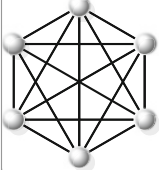
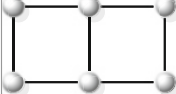
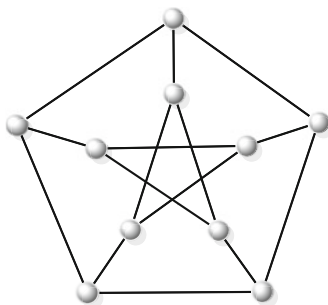
Name	Notation	Vertices	Edges	Example (on 6 vertices)
Path	P_n	n	$n - 1$	
Cycle	C_n	n	n	
Star	S_n	n	$n - 1$	
Wheel	W_n	n	$2(n - 1)$	
Complete graph	K_n	n	$\frac{n(n - 1)}{2}$	
Grid graph	$G_{m,n}$	mn	$(m - 1)n + (n - 1)m$	

Fig. 9 The Petersen graph



graph $G_{m,n}$ is a tree whenever $m = 1$ or $n = 1$, and this happens because then $G_{1,n}$ is a path graph on n vertices and $G_{m,1}$ is a path graph on m vertices.

Graph connectedness is a binary property. In other words, for a graph to be connected, there must exist at least one way to get from any vertex of the graph to any other by traversing (“walking”) along the edges of the graph. If such is not the case for merely one pair of vertices, then the graph is not connected. However, there may actually be multiple ways to do a walk. In light of this, we might want to be as efficient as possible in our “walking” and only traverse the minimum number of edges required to get from vertex u to vertex v . This is the concept of *distance* within a graph, which is defined as the minimum number of edges we can “walk” along to get from vertex u to vertex v . The distance between u and v is denoted by $\text{dist}(u, v)$. By convention, if we cannot find a “walk” that allows us to go from u to v , then $\text{dist}(u, v) = \infty$.

There are many other definitions that allow us to describe the structure of a graph. We begin with the following. Two vertices that are both incident to the same edge are said to be *adjacent* and are often called *neighboring* vertices or simply *neighbors* for short. The *degree* of a vertex v in a simple graph, denoted $\text{deg}(v)$, is the number of vertices in $V(G)$ that are adjacent to v . The *maximum degree* of a graph G , denoted $\Delta(G)$, is the maximum value of $\text{deg}(v)$ taken over all vertices $v \in V(G)$. Alternatively, the minimum such value, denoted $\delta(G)$, is the *minimum degree* of G . For example, the Petersen graph G (in Fig. 9) satisfies that $\text{deg}(v) = 3$ for every vertex v , so that means that $\delta(G) = \Delta(G) = 3$, whereas the star graph S_6 has 5 vertices with degree one and one vertex with degree five, hence $\delta(S_6) = 1$ and $\Delta(S_6) = 5$. In general, $\delta(S_n) = 1$ and $\Delta(S_n) = n - 1$ for $n \geq 1$.

There are also many useful operations that we can perform on graphs. To *delete an edge* from a graph G , we simply remove the edge from $E(G)$. To *delete a vertex* v from a graph G , we remove v from $V(G)$, and we remove from $E(G)$ any edge that is incident to v . If $S \subseteq V(G) \cup E(G)$, the graph obtained from G by removing the vertices and edges in S is denoted by $G - S$. For example, Fig. 10 illustrates taking the complete graph on six vertices and deleting a vertex along with the edges incident to it. The *complement* of a simple graph G , denoted $\bar{G} = K_{|V(G)|} - E(G)$ the graph obtained by deleting all the edges of G from the complete graph $K_{|V(G)|}$.

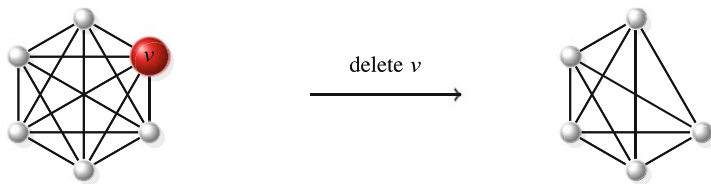


Fig. 10 Deleting the vertex v from K_6

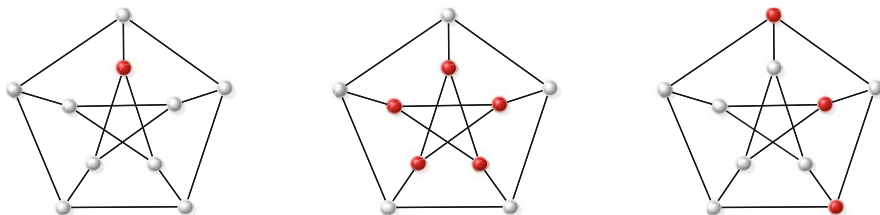


Fig. 11 In each subfigure, S is made up of the colored vertices. Left: a set S , which does not dominate the Petersen graph. Center: a set S , which dominates the Petersen graph. Right: a set S , which dominates the Petersen Graph with a minimum number of vertices



Fig. 12 An example of a digraph, which is an orientation of P_6

Sometimes, we are interested in certain subsets of $V(G)$ with special properties. Given a graph G , a dominating set of G is a set of vertices such that any vertex in the graph is either in the set or adjacent to a vertex in the set.

When S is a dominating set of graph G , we say that S dominates G . Note that for any graph, selecting every vertex of the graph will always lead to a dominating set. We might then ask how much smaller we could make the size of a dominating set. That is the idea behind the *domination number* of a graph. The *domination number* of G , $\gamma(G)$ is the minimum size (order) of a dominating set of G . For example, Fig. 11 illustrates a subset of vertices of the Petersen graph G which does not dominate G , another which does, and a third which does so with a minimum number of vertices possible.

In a *directed graph (or digraph)*, the edges (also called *arcs*) are ordered pairs of vertices. This gives a direction to the edges because in a digraph Γ , it is possible that $(u, v) \in E(\Gamma)$, yet $(v, u) \notin E(\Gamma)$. We can draw the arc (u, v) as an arrow that points from u to v . In this case, u is called an *in-neighbor* of v and v is called an *out-neighbor* of u . In Fig. 12, we orient the edges of the path graph P_6 to create a digraph. If Γ is a digraph and $v \in V(\Gamma)$, the *in-degree* of v , denoted $\deg^-(v)$, is the number of in-neighbors of v in $V(\Gamma)$. Likewise, the *out-degree* of v , denoted $\deg^+(v)$, is the number of out-neighbors of v in $V(\Gamma)$. If we start with a simple graph G and replace each edge $uv \in E(G)$ with either the arc (u, v) or the arc (v, u) , the resulting digraph is called an *orientation* of G . Note that Fig. 12 is an orientation of the path graph P_6 .

References

1. David Blessing, Katie Johnson, Christie Mauretour, and Erik Insko. On (t, r) broadcast domination numbers of grids. *Discrete Appl. Math.*, 187:19–40, 2015.
2. Joshua Carlson, Robin Eagleton, Jesse Geneson, John Petrucci, Carolyn Reinhart, and Preetul Sen. The damage throttling number of a graph. Under review.

3. Danielle Cox and Asiyeh Sanaei. The damage number of a graph. *Australas. J. Combin.*, 75:1–16, 2019.
4. Peter Dankelmann, David Day, David Erwin, Simon Mukwembi, and Henda Swart. Domination with exponential decay. *Discrete Math.*, 309(19):5877–5883, 2009.
5. Pamela E. Harris, Peter Hollander, and Erik Insko. The (t,r) broadcast domination number of directed graphs. in-progress, 2021.
6. Pamela E. Harris, Erik Insko, and Katie Johnson. Projects in (t,r) broadcast domination.
7. Teresa W. Haynes, Stephen T. Hedetniemi, and Peter J. Slater. *Fundamentals of domination in graphs*, volume 208 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, 1998.
8. Richard Nowakowski and Peter Winkler. Vertex-to-vertex pursuit in a graph. *Discrete Math.*, 43(2-3):235–239, 1983.
9. A. Quilliot. Some results about pursuit games on metric spaces obtained through graph theory techniques. *European J. Combin.*, 7(1):55–66, 1986.
10. Mihalis Yannakakis. Edge-deletion problems. *SIAM J. Comput.*, 10(2):297–309, 1981.